# DOUBLE JUMP INVERSIONS AND STRONG MINIMAL COVERS IN THE TURING DEGREES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Yuval Gabay

August 2004

# DOUBLE JUMP INVERSIONS AND STRONG MINIMAL COVERS IN THE TURING DEGREES

Yuval Gabay, Ph.D.

Cornell University 2004

Decidability problems for (fragments of) the theory of the structure $\mathcal{D}$ of Turing degrees, form a wide and interesting class, much of which is yet unsolved. Lachlan showed in 1968 that the first order theory of $\mathcal{D}$ with the Turing reducibility relation is undecidable. Later results concerned the decidability (or undecidability) of fragments of this theory, and of other theories obtained by extending the language (e.g. with $\mathbf{0}$ or with the Turing jump operator). Proofs of these results often hinge on the ability to embed certain classes of structures (lattices, jump-hierarchies, etc.) in certain ways, into the structure of Turing degrees. The first part of the dissertation presents two results which concern embeddings onto initial segments of $\mathcal{D}$ with known double jumps, in other words a double jump inversion of certain degree structures onto initial segments. These results may prove to be useful tools in uncovering decidability results for (fragments of) the theory of the Turing degrees in languages containing the double jump operator.

The second part of the dissertation relates to the problem of characterizing the Turing degrees which have a strong minimal cover, an issue first raised by Spector in 1956. Ishmukhametov solved the problem for the recursively enumerable degrees, by showing that those which have a strong minimal cover are exactly the r.e. weakly recursive degrees. Here we show that this characterization fails outside the r.e. degrees, and also construct a minimal degree below $\mathbf{0}'$ which is not weakly recursive, thereby answering a question from Ishmukhametov's paper.

# BIOGRAPHICAL SKETCH

Yuval Gabay was born in the third person in Jerusalem, Israel. After graduating from the Hebrew University High School in 1992, he served in the Israeli Defense Forces for three years. He then attended the Hebrew University of Jerusalem, graduating with honors in 1998 with a B.Sc. in Mathematics and a B.Sc. in Computer Science. In 2002 he received an M.Sc. in Computer Science from Cornell University. He accepted a post-doctoral offer from the Mathematics Department at the Ben-Gurion University of the Negev in Beer-Sheva, Israel, starting October 2004. He firmly believes that eggplants were never meant to be food.

In memory of my mother.

# ACKNOWLEDGMENTS

One could argue that it is impossible to thank, or even list, all those who have made it possible, indeed inevitable, that I go down the road which has led me to a successful completion of this dissertation. One could say that such a list would have to include every person whom I ever had any contact with, as well as every person whom *they* ever had any contact with (prior to the time of their contact with me), and so on. Had any of these contacts failed to exist in the way that it did, one could explain, the chain of influences may have been irreparably broken. Thus for example, Chapter 4 could be attributed to the police officer who once caught me speeding, Chapter 5 may be claimed by the aunt of my high school janitor, and her cat's veterinarian could be responsible for the Introduction. This paragraph, however, is unmistakably and entirely my fault.

I thank my parents above all. My father infected me with an insatiable curiosity about the inner workings of the universe. My numerous and fascinating childhood visits to his labs at the Hebrew University have made the pleasant grounds of the Givat Ram campus a second home to me. My mother, a philosopher at heart, exposed me to the tremendous joy of thinking. She is responsible for the eagerness with which I have developed my skill of clear and unconstrained thought, a bare necessity of mathematical research. It saddens me that she did not live to see this document and to celebrate her significant part in its authorship. I thank both my sisters for their love and care, and for not letting my being their family get in the way of their being my friends.

I thank my advisor, Richard Shore, whom I cannot thank enough. Infinitely patient, Richard has guided me with calm and confidence through the precarious labyrinth of being a graduate student, having faith in my abilities even when I did not. Richard has been an inspiring teacher, both on the professional level and on the personal one. At this point I still do not know if I want to be a mathematician, but because of Richard I know that I can.

I thank Dexter Kozen and Anil Nerode, members of my special committee. It has been a pleasure to know them and to get a taste of their mathematical interests and abilities. I only wish I had spent more of my time at Cornell interacting with both of them. I thank Juris Hartmanis, whose Complexity Theory course I thoroughly enjoyed first as a student and then (not as thoroughly) as a teaching assistant. The brief conversations we have had were positive and helpful, and I consider his acquaintance a privilege.

Finally, I thank my friends, whom there is no need to name. For you were there, and are still, every step of the way, willing to listen to my ramblings concerning mathematics, my love for it and my frequent frustration with it (your personal preference notwithstanding). Of all mentioned here, you are the only ones of whom I ask to continue performing your seemingly ungratifying role. If not for others, this work would surely have not been written. If not for you it may have been written, but surely not by me.

# Table Of Contents

# Chapter 1

# Introduction

The most significant part of Recursion Theory has always been the study of the structure $\mathcal{D}$ of Turing Degrees. This study has two main interests. One is of algebraic nature, asking local and global questions about the existence of certain formations of degrees. The second interest is logical, concentrating on computable complexity results for the theory (or sub-theories) of $\mathcal{D}$ (or substructures of it), in various languages. These two interests are strongly related. Most decidability results are based on algebraic findings, usually the existence of embeddings of certain structures into $\mathcal{D}$. The earliest such result was Lachlan's proof of the undecidability of $\mathrm{Th}(\mathcal{D}, \leq)$ [Lac68], which is based on the embeddability of any countable distributive lattice as an initial segment of $\mathcal{D}$ (the embeddability of finite distributive lattices suffices, see [Ler83, VI.4.6]). It was shown independently by Shore [Sho78] and Lerman (see [Ler83, VII.4.4]) that $\mathrm{Th}_{\forall\exists}(\mathcal{D}, \leq)$ is decidable. This result is based on the embeddability of any finite lattice as an initial segment of $\mathcal{D}$ and extension of embedding results. Later results concern the decidability of fragments of the theory of $\mathcal{D}$ with added operators, such as lattice operators and the Turing jump (as the jump operator has been recently shown to be definable in $\mathrm{Th}(\mathcal{D}, \leq)$ [SS99], these language expansions affect only the complexity of theory fragments, not that of the whole theory).

This dissertation deals with two issues of the algebraic kind. The first, with an eye toward decidability results for fragments of the theory of $\mathcal{D}$ with languages containing the double jump operator, concerns the embeddability of certain structures as initial segments of $\mathcal{D}$ with prespecified double jumps. The second issue dealt with is the problem of characterizing the Turing degrees which have strong minimal covers.

## 1.1 Double Jump Inversion

While there is relatively little work concerning the inversion specifically of the double jump, much has been said about single jump inversion. The most basic jump inversion theorem is due to Friedberg.

**Theorem 1.1.1.** [Fri57] $\mathbf{a} \geq \mathbf{0}'$ *iff there exists a degree* $\mathbf{c}$ *such that* $\mathbf{c}' = \mathbf{a}$.

Later results, such as the Sacks Inversion [Sac63] or the Cooper Inversion [Coo73] put further requirements on the given degree and on the produced, inverted degree (see Theorems 5.1.1 and 5.1.2). However, the proofs of these theorems are very elaborate, and other similar problems are still in need of a solution. For instance, finite partial orders of degrees above $\mathbf{0}'$ can be jump inverted [Sho88, 1.8], but there is no known general method for inverting lattices (or even upper semi-lattices). The results we bring here attempt to suggest that double jump inversion is much more natural and flexible than its single jump counterpart.

In Chapter 3 we show that certain finite lattices of degrees (distributive ones in particular) above $\mathbf{0}''$ can be double jump inverted to initial segments of $\mathcal{D}$. In Chapter 4 we show a similar result for countable linear orderings recursive in $\mathbf{0}''$ (Simpson [Sim77] has proven this for orders of type $\omega$).

In Chapter 5 we directly (double jump) invert any degree r.e. in and above $\mathbf{0}''$ to a minimal degree below $\mathbf{0}''$. From this it follows that the degrees r.e. in and above $\mathbf{0}''$ are exactly the jumps of minimal degrees below $\mathbf{0}'$. To contrast, not all degrees r.e. in and above $\mathbf{0}'$ are jumps of minimal degrees below $\mathbf{0}'$, as all such jumps must be low over $\mathbf{0}'$ (that is, having $\mathbf{0}''$ as their jump). Moreover, even the degrees r.e. in and low over $\mathbf{0}'$ cannot all be inverted to minimal degrees below $\mathbf{0}'$ [DLS96].

We conclude this part by showing, as a corollary of Shore's Non-Inversion Theorem [Sho88, 1.2], that this inversion result for degrees r.e. in and above $\mathbf{0}''$ cannot be extended for even the simplest lattices.

## 1.2 Strong Minimal Covers

A degree $\mathbf{b}$ is a strong minimal cover of a degree $\mathbf{a}$ below it, if every degree $< \mathbf{b}$ is $\leq \mathbf{a}$. In 1956 Spector constructed a minimal degree, which is by nature a strong minimal cover of $\mathbf{0}$, and raised the issue of characterizing those degrees which possess a strong minimal cover [Spe56]. One should note that relativizing the minimal degree construction to $\mathcal{D}(\geq \mathbf{a})$ gives only a minimal cover of $\mathbf{a}$ (i.e. a degree $\mathbf{b} > \mathbf{a}$ with no degrees strictly between the two).

Several constructions of degrees with strong minimal covers were presented over the years. The first serious stab at a characterization was made by Downey, Jockush and Stob. In [DJS96a] they defined the notion of r.e. array non-recursive degrees, and later extended it to $\mathcal{D}$ in general [DJS96b].

**Definition 1.2.1.** A degree $\mathbf{a}$ is *array non-recursive* if for each $f \leq_{\mathrm{wtt}} K$ there is a function $g$ recursive in $\mathbf{a}$ such that $g(n) \geq f(n)$ for infinitely many $n$.

In other words, $\mathbf{a}$ is a.n.r. if no function weak-truth-table recursive in $\mathbf{0}'$ dominates all $g \leq_T \mathbf{a}$. This clearly implies that the class of a.n.r. degrees is upward closed. This definition has other interesting equivalents which we will not go into here.

**Theorem 1.2.2.** [DJS96b] *Let* $\mathbf{a}$ *be array non-recursive.*

*(i)* $\mathbf{a}$ *is the supremum of two 1-generic a.n.r. degrees which form a minimal pair.*

*(ii) If* $\mathbf{c} > \mathbf{a}$ *then there is a degree* $\mathbf{b} < \mathbf{c}$ *such that* $\mathbf{a} \vee \mathbf{b} = \mathbf{c}$.

By (i), a.n.r. degrees cannot be strong minimal covers. By (ii), they cannot have strong minimal covers (alternatively, one could argue that a strong minimal cover of an a.n.r. degree would be a.n.r. by upward closure, thereby contradicting (i)).

The second half of the characterization came, for the r.e. degrees, in a 1999 paper by Ishmukhametov [Ish99]. He defined the class of weakly recursive degrees (see Definition 6.1.2), argued that it is complementary to the class of a.n.r. degrees in the r.e. degrees, and then proved that all weakly recursive degrees possess strong minimal covers, concluding

**Theorem 1.2.3.** *If* $\mathbf{a}$ *is recursively enumerable, it has a strong minimal cover iff it is weakly recursive.*

In Chapter 6 we construct a minimal degree which is not weakly recursive, but has a strong minimal cover, showing that the characterization in Theorem 1.2.3 does not hold outside the realm of r.e. degrees. This is also done independently in [KHNS] (where the weakly recursive degrees are called *r.e.-traceable*).

In Chapter 7 we construct a minimal degree below $\mathbf{0}'$ which is not weakly recursive, thereby showing that the a.n.r. and weakly recursive classes are not complementary in the degrees below $\mathbf{0}'$ (note that a minimal degree is not a.n.r. by 1.2.2(i)). This answers an open question in [Ish99].

In Chapter 8 we define the seemingly broader class of *very weakly recursive degrees*, and show that those possess a strong minimal cover as well, by a slight modification of Ishmukhametov's argument. At this time we do not know if there are very weakly recursive degrees which are not weakly recursive.

Chapter 9 has no new results. It describes a direct construction of a degree **a** for which the statement 1.2.2(ii) does not hold. This is a product of a failed attempt to produce a counterpart to Theorem 6.2.1, by constructing a degree which is not a.n.r., yet does not have a strong minimal cover. Nonetheless, we have chosen to include it, since we feel it may prove useful in the future, either in this or in other research routes, such as the characterization of the class of cuppable degrees (i.e. degrees for which 1.2.2(ii) does hold).

# Chapter 2

# Basic Definitions and Notation

## 2.1 Recursion Theory

The $e$-th partial recursive function with oracle $A \subseteq \omega$ will be denoted $\varphi_e^A$. The same function, computed for only $s$ steps, will be denoted $\varphi_{e,s}^A$. We write $\varphi_e^A(x)\downarrow$ or $\varphi_e^A(x)\uparrow$ to mean the function is defined or undefined on an input $x \in \omega$, respectively (and similarly for $\varphi_{e,s}^A$).

We write $\varphi_e^A \simeq \varphi_i^A$ to mean

$$\forall x \in \omega\ \varphi_e^A(x)\downarrow = \varphi_i^A(x) \vee (\varphi_e^A(x)\uparrow \wedge \varphi_i^A(x)\uparrow).$$

We write $\varphi_e^A = B$ to mean that for all $x \in \omega$, $(x \notin B \to \varphi_e^A(x)\downarrow = 0) \wedge (x \in B \to \varphi_e^A(x)\downarrow = 1)$.

**Definition 2.1.1.** $A$ is *(Turing) computable from $B$* if $\varphi_e^A = B$ for some $e \in \omega$. In this case we write $A \leq_T B$.

The relation $A \equiv_T B$ defined as $A \leq_T B \wedge B \leq_T A$ is an equivalence relation on $2^\omega$, which divides it into countable equivalence classes, the *Turing degrees.* The set of Turing degrees is denoted by $\mathcal{D}$, and individual degrees by boldface letters $(\mathbf{a}, \mathbf{b}, \dots)$. The Turing degree of a set $A$ is denoted $\deg(A)$. The least degree is the set of all recursive subsets of $\omega$, denoted $\mathbf{0}$. We write $A <_T B$ to mean that $A$ is computable from $B$ and they are of distinct Turing degrees.

**Definition 2.1.2.** For $A \subseteq \omega$, the *Turing jump of* $A$ is the set

$$A' = \{e : \varphi_e^A(e)\downarrow\}$$

$A <_T A'$ and if $A \leq_T B$ then $A' \leq_T B'$. Consequently the Turing jump is well defined on Turing degrees, which allows us to define the jump of a Turing degree $\mathbf{a} = \deg(A)$ as $\mathbf{a}' = \deg(A')$.

The following facts will be used throughout the text, sometimes implicitly.

**Theorem 2.1.3.** [Ler83, III.2.6] *For $A \subseteq \omega$ we have*

- *$A \in \Sigma^0_{n+1}$ iff $A$ is recursively enumerable in $\mathbf{0}^{(n)}$.*
- *$A \in \Delta^0_{n+1}$ iff $A$ is recursive in $\mathbf{0}^{(n)}$.*

Hence, if $A$ is definable by a formula $\psi(x)$ in normal form with $n$ quantifiers, it is computable from $\mathbf{0}^{(n)}$.

**Lemma 2.1.4.** [Ler83, IV.3.2] *The* totality set *of $A$,*

$$\mathrm{Tot}(A) = \{e : \varphi_e^A \textit{ is total}\},$$

*is Turing equivalent to $A''$.*

**Lemma 2.1.5.** [Ler83, IV.3.6] *Any degree $\mathbf{a}$ in a finite initial segment of $\mathcal{D}$ is in $\mathbf{GL}_2$, that is $\mathbf{a}'' = (\mathbf{a} \vee \mathbf{0}')'$.*

We denote by $\langle \cdot, \cdot \rangle$ a fixed recursive bijection between $\omega \times \omega$ and $\omega$. $\langle x, y, z \rangle$ is used to mean $\langle x, \langle y, z \rangle \rangle$, and so on.

The *set-join* of $A, B \in \omega$ is the set

$$A \oplus B = \{\langle i, x\rangle \,:\, (i = 0 \wedge x \in A) \vee (i = 1 \wedge x \in B)\}$$

If $A \leq_T C$ and $B \leq_T D$ then $A \oplus B \leq_T C \oplus D$. This allows us to define the join of two Turing degrees $\mathbf{a} = \deg(A)$ and $\mathbf{b} = \deg(B)$ as $\mathbf{a} \vee \mathbf{b} = \deg(A \oplus B)$. This is the least upper bound of the degrees $\mathbf{a}$ and $\mathbf{b}$.

## 2.2 Strings and Trees

Let $\mathcal{B}$ be a non-empty at most countable set.

**Definition 2.2.1.** A $\mathcal{B}$-*string* is a member of $\mathcal{B}^{\leq\omega}$. For a finite $\mathcal{B}$-string $\sigma$ and a $\mathcal{B}$-string $\tau$ we define:

(i) The *length* of $\tau$ is $|\tau|$. We identify elements of $\mathcal{B}$ and strings of length 1.

(ii) If $\sigma \subseteq \tau$ (as functions), we say that $\sigma$ is an *initial segment* of $\tau$.

(iii) The *concatenation* $\sigma \frown \tau$ is the string $\rho$ of length $|\sigma| + |\tau|$ defined by

$$\rho(n) = \begin{cases} \sigma(n) & n < |\sigma| \\ \tau(n - |\sigma|) & |\sigma| \leq n < |\sigma| + |\tau| \end{cases}$$

(iv) If $\sigma \subseteq \tau$ let $\tau - \sigma$ denote the unique $\rho$ such that $\sigma \frown \rho = \tau$.

(v) If $\sigma \subseteq \tau$ and $|\rho| = |\sigma|$, we denote $\rho \frown (\tau - \sigma)$ by the special notation $\tau[\sigma \to \rho]$.

We naturally identify a binary string (i.e. a $\{0,1\}$-string) $A$ with the set $A^{-1}(1)$.

**Definition 2.2.2.** Let $A$ be a $\mathcal{B}$-string and let $X \subseteq \omega$ be infinite and enumerated by $m_0, m_1, m_2, \ldots$ in increasing order. $X(A)$ is the $\mathcal{B}$-string $C$ such that $C(n) =$

$A(m_n)$ for all $n$ for which this is defined. If $X(A_1)$ is compatible with $X(A_2)$, we say that $A_1$ and $A_2$ *agree on* $X$.

Note that $X(A)$ is infinite iff $A$ is. Also, if $A$ is infinite, $X, Y$ are recursive and $X \subseteq^* Y$ (that is, $X \setminus Y$ is finite) then $X(A) \leq_T Y(A)$. In particular, $X(A) \leq_T \omega(A) = A$ for all $X$.

**Definition 2.2.3.** Let $\sigma, \tau$ be (finite or infinite) $\mathcal{B}^k$-strings (for $k > 0$), and $i < k$.

(i) $\sigma^{[i]}$ is the $\mathcal{B}$-string given by $\sigma^{[i]}(n) = \sigma(n)(i)$.

(ii) $\sigma$ and $\tau$ are *equivalent mod* $i$ ($\sigma \equiv_i \tau$) if $\sigma^{[i]} = \tau^{[i]}$.

**Definition 2.2.4.** (i) A $\mathcal{B}$-*tree* is an injective function $T : \mathcal{B}^{<\omega} \to \mathcal{B}^{<\omega}$ such that

$$\sigma \subseteq \tau \leftrightarrow T(\sigma) \subseteq T(\tau)$$

(ii) Given a $\mathcal{B}$-tree $T$ and an infinite $\mathcal{B}$-string $A$, we can define the infinite $\mathcal{B}$-string $P = \bigcup_{\sigma \subset A} T(\sigma)$. $P$ is called a *path on* $T$, and is sometimes denoted by $T(A)$ (slightly abusing notation). The set of paths on $T$ is denoted by $[T]$.

(iii) A $\mathcal{B}$-tree $T$ is said to *force* a property $\mathcal{P}$ of infinite $\mathcal{B}$-strings if every path on $T$ satisfies $\mathcal{P}$. A finite string $\sigma \in \mathrm{rg}(T)$ forces $\mathcal{P}$ if every path extending $\sigma$ on $T$ satisfies $\mathcal{P}$.

**Definition 2.2.5.** For $\mathcal{B}$-trees $S$ and $T$, $S$ is a *subtree* of $T$ ($S \subseteq T$) if the range of $S$ is contained in the range of $T$. Note that in this case $[S] \subseteq [T]$ as well.

Let $S, T$ be $\mathcal{B}$-trees. The composition $R = T \circ S$ is injective, and for all $\sigma, \tau \in \mathcal{B}^{<\omega}$ we have

$$\sigma \subseteq \tau \iff S(\sigma) \subseteq S(\tau) \iff R(\sigma) = T(S(\sigma)) \subseteq T(S(\tau)) = R(\tau).$$

Hence $R$ is a $\mathcal{B}$-tree, and is called the *product* of $T$ and $S$ (denoted $T \cdot S$ or simply $TS$).

If $S \subseteq T$, the composition $R = T^{-1} \circ S$ is well defined and injective, and for all $\sigma, \tau \in \mathcal{B}^{<\omega}$ we have

$$\sigma \subseteq \tau \quad \Longleftrightarrow \quad T(R(\sigma)) = S(\sigma) \subseteq S(\tau) = T(R(\tau)) \quad \Longleftrightarrow \quad R(\sigma) \subseteq R(\tau).$$

Hence $R$ is a $\mathcal{B}$-tree, and is called the *quotient* of $S$ by $T$ (denoted $S/T$).

**Proposition 2.2.6.** *Let $R, S, T$ be $\mathcal{B}$-trees. $T_{id}$ denotes the identity $\mathcal{B}$-tree.*

*(i)* $T \cdot (S \cdot R) = (T \cdot S) \cdot R$.

*(ii)* $T/T = T_{id}$ *and* $T/T_{id} = T_{id} \cdot T = T \cdot T_{id} = T$.

*(iii)* *If* $S \subseteq T$ *then* $T \cdot (S/T) = S$.

*(iv)* *If* $S \subseteq T$ *then* $S \cdot R \subseteq T$, *and* $(S \cdot R)/T = (S/T) \cdot R$.

*(v)* *If* $T \subseteq R \cdot S$ *then* $T \subseteq R$, $T/R \subseteq S$ *and* $(T/R)/S = T/(R \cdot S)$.

*(vi)* *If* $S \subseteq T$ *then* $R \cdot S \subseteq R \cdot T$ *and* $(R \cdot S)/(R \cdot T) = S/T$.

*Proof.* Immediate from the definitions. $\square$

For a finite $\mathcal{B}$-string $\sigma$, let $E_\sigma$ be the tree defined by $E_\sigma(\tau) = \sigma \frown \tau$. If $T$ is a $\mathcal{B}$-tree then $T \cdot E_\sigma$ is an *extension subtree* of $T$ (denoted $\mathrm{Ext}(T, \sigma)$ in Lerman).

Since $\mathcal{B}$ is at most countable, we allow ourselves to identify $\mathcal{B}$-strings with their codes (in some fixed Gödel numbering). In this light, a tree $T$ is a numerical function, and has a Turing degree.

Furthermore, when we talk about operators on recursive trees, we implicitly (and sometimes explicitly) refer to corresponding functions on recursive indices of trees (even though these are not unique). For example, there is a (recursive!)

function $f.(s,t)$ which maps a pair of recursive indices for trees $S, T$ to a recursive index for $S \cdot T$. We will frequently write $S \cdot T$ to mean $f.(s,t)$.

Finally we address an important application of these tree operations.

**Definition 2.2.7.** Two $\mathcal{B}$-tree extensions $T' \subseteq T$ and $S' \subseteq S$ are *equivalent* if $T'/T = S'/S$.

Informally speaking, the extensions are equivalent if $T'(\sigma)$ is on the same place on $T$ as $S'(\sigma)$ is on $S$ (for all strings $\sigma$). The tree operations let us construct equivalent extensions with ease.

**Proposition 2.2.8.** *If $T' \subseteq T$ is a $\mathcal{B}$-tree extension and $S$ is a $\mathcal{B}$-tree, then the tree*

$$S' = S \cdot (T'/T)$$

*is an extension of $S$, equivalent to $T' \subseteq T$.*

*Proof.* By Proposition 2.2.6 (v) we have that $S' \subseteq S$ and

$$(S'/S)/(T'/T) = S'/(S \cdot (T'/T)) = S'/S' = T_{\mathrm{id}}.$$

$\square$

# Chapter 3

# Finite Distributive Lattices

In this chapter we focus on embedding certain lattices as initial segments with given double jumps. Most of the introductory material is taken from Chapter IV of [Ler83]. In section 2 we present the main construction, and in section 3 we prove the results that are based on it.

## 3.1 Introduction

In the following, $\mathcal{L} = (L, \leq, \vee, \wedge)$ is a finite lattice, where $L = \{u_0, \ldots, u_n\}$, and the distinct bottom and top elements of $\mathcal{L}$ are $u_0$ and $u_n$, respectively. Also, $h : \mathcal{L} \to \mathcal{D}$ is an u.s.l. homomorphism (i.e. a map preserving the lattice weak order relation and the join function).

**Definition 3.1.1.** A *finite homogeneous lattice table* for $\mathcal{L}$ is a set $\Theta$ of functions from $\{0, \ldots, n\}$ to $\omega$, such that

(i) $\forall \alpha, \beta \in \Theta (\alpha \equiv_0 \beta)$ (equivalence mod 0 of $\Theta$-strings of length 1)

(ii) $\forall i, j \leq n (u_i \leq u_j \leftrightarrow \forall \alpha, \beta \in \Theta (\alpha \equiv_j \beta \to \alpha \equiv_i \beta))$

(iii) $\forall i, j, k \leq n (u_i \vee u_j = u_k \leftrightarrow \forall \alpha, \beta \in \Theta (\alpha \equiv_k \beta \leftrightarrow \alpha \equiv_i \beta \wedge \alpha \equiv_j \beta))$

(iv) $\forall i, j, k \leq n(u_i \wedge u_j = u_k \leftrightarrow \forall \alpha, \beta \in \Theta(\alpha \equiv_k \beta \leftrightarrow$
$\exists s \in \omega\, \exists \gamma_0, \ldots, \gamma_s \in \Theta(\alpha = \gamma_0 \equiv_i \gamma_1 \equiv_j \gamma_2 \equiv_i \cdots \equiv_j \gamma_s = \beta)))$

(v) For all $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \Theta$, if $\forall i \leq n(\alpha_0 \equiv_i \alpha_1 \rightarrow \beta_0 \equiv_i \beta_1)$, then there is a function $f : \Theta \rightarrow \Theta$ such that $f(\alpha_0) = \beta_0$, $f(\alpha_1) = \beta_1$, and for all $\alpha, \beta \in \Theta$, $\forall i \leq n(\alpha \equiv_i \beta \rightarrow f(\alpha) \equiv_i f(\beta))$.

We are concerned here only with finite lattices which have a finite homogeneous lattice table. In the following, assume that $\Theta$ is such a table for $\mathcal{L}$. Embeddings of $\mathcal{L}$ into $\mathcal{D}$ will be given by infinite $\Theta$-strings.

**Definition 3.1.2.** An infinite $\Theta$-string $A$ *codes* a map $g : \mathcal{L} \rightarrow \mathcal{D}$ if for all $i \leq n$

$$g(u_i) = \deg(A^{[i]}) \vee g(u_0).$$

A map $g : \mathcal{L} \rightarrow \mathcal{D}$ is $\Theta$-*representable* if there exists a $\Theta$-string that codes it.

Every $\Theta$-string codes a u.s.l. homomorphism $g$ with $g(u_0) = \mathbf{0}$ (this is essentially the Homomorphism Lemma, [Ler83, IV.1.4]). In order to make $g$ an embedding onto an initial segment, we construct the string as a path on uniform $\Theta$-trees forcing certain conditions.

**Definition 3.1.3.** A $\Theta$-tree $T$ is *uniform* if

(i) $|\sigma| = |\tau| \leftrightarrow |T(\sigma)| = |T(\tau)|$

(ii) $\sigma \equiv_i \tau \leftrightarrow T(\sigma) \equiv_i T(\tau)$

(iii) $|\sigma| = |\tau| \rightarrow \forall \alpha \in \Theta\ T(\sigma \frown \alpha) - T(\sigma) = T(\tau \frown \alpha) - T(\tau)$

Note that condition (iii) implies

(iii') $|\sigma| = |\tau| \rightarrow \forall \rho \in \Theta^{<\omega}\ T(\sigma \frown \rho) - T(\sigma) = T(\tau \frown \rho) - T(\tau)$

by induction on the length of $\rho$. Also note that if $S, T$ are recursive uniform trees then so are $S \cdot T$ and $S/T$ (when defined).

**Definition 3.1.4.** Let $e \in \omega$, $i, j \leq n$ and let $T$ be a $\Theta$-tree.

(i) $T$ *decides $e$-totality* if either $\varphi_e^P$ is total for all $P \in [T]$ or $\varphi_e^P$ is not total for all $P \in [T]$. $T$ *forces $e$-totality, $e$-partiality* in the first, resp. second, case.

(ii) $T$ is *$\langle e, i, j\rangle$-differentiating* if for all $P \in [T]$, $\varphi_e^{P^{[j]}} \neq P^{[i]}$.

(iii) $T$ is *$e$-splitting* if there exists $i \leq n$ such that for all $P \in [T]$ with $\varphi_e^P$ total, we have $\varphi_e^P \equiv_T P^{[i]}$.

Clause (iii) differs from the definition of $e$-splitting in [Ler83], but is shown to follow from it by the Computation Lemma [Ler83, IV.3.3]. Hence, the justification of the following Lemma is not affected.

**Lemma 3.1.5.** *There is a total function $f : \omega \times \omega \to \omega$, computable in $\mathbf{0}''$, such that if $T$ is (a recursive index for) a uniform $\Theta$-tree then*

*(i) if $r = \langle 1, e\rangle$ then $f(r, T)$ is (a recursive index for) a uniform subtree of $T$ which decides $e$-totality;*

*(ii) if $u_i \not\leq u_j$ and $r = \langle 2, e, i, j\rangle$ then $f(r, T)$ is a uniform subtree of $T$ which is $\langle e, i, j\rangle$-differentiating;*

*(iii) if $r = \langle 3, e\rangle$ then $f(r, T)$ is a uniform subtree of $T$ which is $e$-splitting;*

*(iv) for all other values of $r$, $f(r, T) = T$.*

*Proof.* The constructions needed to define $f$ are described in the proofs of Lemmas 2.13, 2.15 and 3.12 in [Ler83, IV]. One should note that the constructions are uniform in $\mathbf{0}''$, as the different cases in each are defined by two quantifier statements. $\square$

## 3.2 The Main Construction

In order to get an embedding of $\mathcal{L}$ onto an initial segment of $\mathcal{D}$, Lerman applies $f$ of Lemma 3.1.5 repeatedly to force all of the totality, differentiating and minimality requirements. The goal in this chapter is to code certain degrees into this construction, in a way which would make these degrees the double jumps of the degrees in the initial segment. Rather than explicitly coding any given degrees, we perform every possible coding simultaneously, by branching into separate forcing extensions after each application of $f$. This creates a tree of trees, as stated by the following Lemma.

**Lemma 3.2.1.** *There is a uniform $\Theta$-tree $\hat{T}$, computable in $\mathbf{0}''$, such that every $P \in [\hat{T}]$ codes an embedding of $\mathcal{L}$ onto an initial segment of $\mathcal{D}$. Furthermore, for any such path $P = \hat{T}(A)$ and any $i \leq n$*

$$P^{[i]''} \equiv_T P^{[i]} \vee \mathbf{0}'' \equiv_T A^{[i]} \vee \mathbf{0}''$$

*Proof.* Let $f$ be as in Lemma 3.1.5. We define recursive uniform $\Theta$-trees $T_\sigma$ for every finite $\Theta$-string $\sigma$, such that

(i) $T_{\sigma\frown\alpha} \subseteq T_\sigma E_\alpha$

(ii) $T_{\sigma\frown\alpha} \subseteq f(|\sigma|, T)$ for some recursive tree $T$

(iii) if $|\sigma_1| = |\sigma_2|$ and $|\rho_1| = |\rho_2|$ then the strings $\chi_d = T_{\sigma_d}(\rho_d)$ $(d = 1, 2)$ are of equal length, and for all $i \leq n$

$$\chi_1 \equiv_i \chi_2 \iff \sigma_1 \frown \rho_1 \equiv_i \sigma_2 \frown \rho_2$$

(iv) if $|\sigma_1| = |\sigma_2|$ then $T_{\sigma_1}(\rho \frown \rho') - T_{\sigma_1}(\rho) = T_{\sigma_2}(\rho \frown \rho') - T_{\sigma_2}(\rho)$.

Each tree $T_\sigma$ is the forcing condition which is obtained by coding $\sigma$ into Lerman's original construction. Property (i) ensures that these conditions are distinct, while property (ii) ensures that the next application of $f$ has taken place. Conditions (iii) and (iv) provide a kind of uniformity for $\hat{T}$, which will make the coding work.

Let $T_\emptyset = T_{\mathrm{id}}$. Suppose $T_\sigma$ has been defined for all $\sigma$ of length $k$. Let $\{\tau_j\}_{j=1}^m$ list all $\Theta$-strings of length $k+1$, and let $\tau_j = \sigma_j \frown \alpha_j$ with $|\sigma_j| = k$ and $\alpha_j \in \Theta$. Let $T'_{\tau_j} = T_{\sigma_j} E_{\alpha_j}$. We go by subinduction on $j$:

$$T^*_{\tau_1} = f(k, T'_{\tau_1}), \qquad T^*_{\tau_{j+1}} = f(k, T'_{\tau_{j+1}}(T^*_{\tau_j}/T'_{\tau_j})).$$

For all $1 \leq j \leq m$ now let

$$T_{\tau_j} = T'_{\tau_j}(T^*_{\tau_m}/T'_{\tau_m}).$$

The idea is simple and rather straightforward. We want to find conditions $T_{\tau_j}$ which force the current requirement, and are all equivalent extensions (in the sense of Definition 2.2.7) of the conditions $T'_{\tau_j}$, respectively. To do this, we apply $f$ to the first one, then find the equivalent extension to the second and apply $f$ again, and so on. Finally, we take the final extension, and set the trees $T_{\tau_j}$ to be the corresponding extensions equivalent to it.

We now formally prove that these extensions are as required. The trees are all uniform recursive, since this quality is preserved by $f$ and by tree multiplication and division. We now show that the properties above hold by induction, assuming the notation of the construction.

(i) This follows directly from the definition, since for all $1 \leq j \leq m$ we have

$$T_{\sigma_j \frown \alpha_j} = T_{\tau_j} = T'_{\tau_j}(T^*_{\tau_m}/T'_{\tau_m}) \subseteq T'_{\tau_j} = T_{\sigma_j} E_{\alpha_j}.$$

(ii) From the definition of $f$ and $T^*_{\tau_{j+1}}$ it follows that

$$T^*_{\tau_{j+1}} \subseteq T'_{\tau_{j+1}}(T^*_{\tau_j}/T'_{\tau_j})$$

and therefore

$$T^*_{\tau_{j+1}}/T'_{\tau_{j+1}} \subseteq T^*_{\tau_j}/T'_{\tau_j}.$$

Hence the quotients $T^*_{\tau_j}/T'_{\tau_j}$ form a decreasing chain of trees. For any $1 \leq j \leq m$ we then get

$$T_{\tau_j} = T'_{\tau_j}(T^*_{\tau_m}/T'_{\tau_m}) \subseteq T'_{\tau_j}(T^*_{\tau_j}/T'_{\tau_j}) = T^*_{\tau_j} = f(k, T)$$

for some recursive $T$ as required.

(iii) Consider strings $\rho_1, \rho_2$ of equal length, and $1 \leq j, l \leq m$. If we let $R = T^*_{\tau_m}/T'_{\tau_m}$, then

$$T_{\tau_j}(\rho_1) = T'_{\tau_j} R(\rho_1) = T_{\sigma_j} E_{\alpha_j} R(\rho_1) = T_{\sigma_j}(\alpha_j \frown R(\rho_1))$$

and similarly

$$T_{\tau_l}(\rho_2) = T_{\sigma_l}(\alpha_l \frown R(\rho_2)).$$

Since $R$ is uniform, we have

$$|\alpha_j \frown R(\rho_1)| = 1 + |R(\rho_1)| = 1 + |R(\rho_2)| = |\alpha_l \frown R(\rho_2)|$$
$$\Longrightarrow \quad |T_{\sigma_j}(\alpha_j \frown R(\rho_1))| = |T_{\sigma_l}(\alpha_l \frown R(\rho_2))|$$

by the induction hypothesis. By a similar reasoning

$$\begin{aligned} T_{\tau_j}(\rho_1) \equiv_i T_{\tau_l}(\rho_2) \quad &\Longleftrightarrow \quad T_{\sigma_j}(\alpha_j \frown R(\rho_1)) \equiv_i T_{\sigma_l}(\alpha_l \frown R(\rho_2)) \\ &\Longleftrightarrow \quad \sigma_j \frown \alpha_j \frown R(\rho_1) \equiv_i \sigma_l \frown \alpha_l \frown R(\rho_2) \\ &\Longleftrightarrow \quad \tau_j \frown \rho_1 \equiv_i \tau_l \frown \rho_2 \end{aligned}$$

(since $R(\rho_1) \equiv_i R(\rho_2)$ iff $\rho_1 \equiv_i \rho_2$ by uniformity of $R$).

(iv) Let $R, j, l$ be as above, and set

$$\chi = R(\rho), \quad \chi' = R(\rho \frown \rho') - R(\rho).$$

Then

$$\begin{aligned} &T_{\tau_j}(\rho \frown \rho') - T_{\tau_j}(\rho) \\ &\quad = T_{\sigma_j}(\alpha_j \frown R(\rho \frown \rho')) - T_{\sigma_j}(\alpha_j \frown R(\rho)) \\ &\quad = T_{\sigma_j}(\alpha_j \frown \chi \frown \chi') - T_{\sigma_j}(\alpha_j \frown \chi) \\ &\quad = T_{\sigma_j}(\alpha_l \frown \chi \frown \chi') - T_{\sigma_j}(\alpha_l \frown \chi) \quad \text{(by uniformity of } T_{\sigma_j}\text{, cond. (iii'))} \\ &\quad = T_{\sigma_l}(\alpha_l \frown \chi \frown \chi') - T_{\sigma_l}(\alpha_l \frown \chi) \quad \text{(by induction hypothesis)} \\ &\quad = T_{\tau_l}(\rho \frown \rho') - T_{\tau_l}(\rho). \end{aligned}$$

Now let $\hat{T}(\sigma) = T_\sigma(\emptyset)$ for all $\sigma$. Property (iii) implies that if $|\sigma| = |\tau|$ then $|\hat{T}(\sigma)| = |\hat{T}(\tau)|$ and for all $i \leq n$

$$\hat{T}(\sigma) \equiv_i \hat{T}(\tau) \leftrightarrow \sigma \equiv_i \tau.$$

Property (iv) implies that if $|\sigma| = |\tau|$ and $\alpha \in \Theta$ then

$$\hat{T}(\sigma \frown \alpha) - \hat{T}(\sigma) = T_\sigma(\alpha \frown \rho) - T_\sigma(\emptyset) = T_\tau(\alpha \frown \rho) - T_\tau(\emptyset) = \hat{T}(\tau \frown \alpha) - \hat{T}(\tau)$$

where $\rho = R(\emptyset)$ with $R$ as above. Therefore $\hat{T}$ is a uniform tree. It is computed with $f$ as an oracle, hence it is computable in $\mathbf{0}''$.

Fix $P = \hat{T}(A) \in [\hat{T}]$. We show that $P$ codes an embedding of $\mathcal{L}$ onto an initial segment of $\mathcal{D}$. First, note that $P \in [T_\sigma]$ whenever $\sigma \subset A$. Indeed,

$$P = \bigcup_{\sigma \subseteq \tau \subset A} \hat{T}(\tau) = \bigcup_{\sigma \subseteq \tau \subset A} T_\tau(\emptyset)$$

and $T_\tau \subseteq T_\sigma$ for every $\tau \supseteq \sigma$ (by property (i)).

From property (ii) it now follows that for every $k \in \omega$ there is a recursive tree $T$ such that $P \in [f(k, T)]$. In particular

(i) If $u_i \not\leq u_j$ then for every $e \in \omega$, $P$ lies on a recursive $\langle e, i, j\rangle$-differentiating tree, and therefore $P^{[i]} \not\leq_T P^{[j]}$.

(ii) For all $e \in \omega$, $P$ lies on a recursive $e$-splitting tree, and therefore

$$\mathcal{D}(\leq \deg(P)) = \mathcal{D}(\leq \deg(P^{[n]})) \subseteq \{P^{[i]} \,:\, i \leq n\}.$$

Thus $P$ codes an embedding of $\mathcal{L}$ onto an initial segment of $\mathcal{D}$. It is left to prove the triple equality in the Lemma. Since $\mathbf{a}'' \geq_T \mathbf{a} \vee \mathbf{0}''$ for any degree $\mathbf{a}$, it suffices to demonstrate that the inequalities

$$P^{[i]} \vee \mathbf{0}'' \geq_T A^{[i]} \vee \mathbf{0}'' \geq_T P^{[i]''}$$

hold for every $i \leq n$. The first of the two follows from the uniformity of $\hat{T}$. If $\sigma$ is a $\Theta$-string of length $k$, then

$$\begin{aligned} \sigma^{[i]} \subseteq A^{[i]} \quad &\iff \quad \sigma \equiv_i A \restriction k \\ &\iff \quad \hat{T}(\sigma) \equiv_i \hat{T}(A \restriction k) \quad \iff \quad \hat{T}(\sigma)^{[i]} \subseteq P^{[i]}. \end{aligned}$$

Hence, to find $A^{[i]} \restriction k$ (with the given oracle) look for a string $\sigma$ of length $k$ which satisfies the last statement, and take $\sigma^{[i]}$.

For the second inequality, we show how to compute $\mathrm{Tot}^{P^{[i]}}$ using $A^{[i]} \vee \mathbf{0}''$ as an oracle (this suffices by Lemma 2.1.4). Given $e \in \omega$, let $e^*$ be such that $\varphi^P_{e^*} \simeq \varphi^{P^{[i]}}_e$ ($e^*$ can be found in a uniformly recursive manner). Let $\sigma$ be of length $k = \langle 1, e^* \rangle + 1$ such that $\sigma^{[i]} \subset A^{[i]}$. Note that both $T_\sigma$ and $T_{A|k}$, decide $e^*$-totality by property (ii) and the nature of the function $f$. Now

$$\begin{aligned} e \in \mathrm{Tot}^{P^{[i]}} \quad &\iff \quad \varphi^{P^{[i]}}_e \text{ is total} \\ &\iff \quad \varphi^P_{e^*} \text{ is total} \\ &\iff \quad T_{A|k} \text{ forces } e^*\text{-totality} \quad \iff \quad T_\sigma \text{ forces } e^*\text{-totality} \end{aligned}$$

where the last equivalence stems from the equality

$$\{X^{[i]} \,:\, X \in [T_{A|k}]\} = \{X^{[i]} \,:\, X \in [T_\sigma]\}$$

which is a consequence of property (iv) and the equivalence $T_{A|k} \equiv_i \sigma$. $\square$

## 3.3 The Results

Each path of $\hat{T}$ codes an embedding of $\mathcal{L}$ onto an initial segment of $\mathcal{D}$. Before we pay attention to the coding in the paths, we mention their cardinality.

**Corollary 3.3.1.** *If $\mathcal{L}$ has a finite homogeneous lattice table, then there are $2^\omega$ many lattice embeddings of $\mathcal{L}$ as an initial segment of $\mathcal{D}$.*

*Proof.* For each path $P \in [\hat{T}]$, the function $h_P : u_i \mapsto \deg(P^{[i]})$ is such an embedding. Now

$$2^\omega \geq |\{h_P : P \in [\hat{T}]\} \geq |\{\deg(P^{[n]}) : P \in [\hat{T}]\} = |\{\deg(P) : P \in [\hat{T}]\}| = 2^\omega$$

since $|[\hat{T}]| = 2^\omega$ and Turing equivalence classes are countable. □

Now we turn to the main Theorem of this chapter, in its most general form.

**Theorem 3.3.2.** *Suppose $\mathcal{L}$ has a finite homogeneous lattice table $\Theta$, and $h : \mathcal{L} \to \mathcal{D}$ is a $\Theta$-representable u.s.l. homomorphism with $h(u_0) = \mathbf{0}''$. Then there is an initial segment $\mathcal{C}$ of $\mathcal{D}$ and a lattice isomorphism $g : \mathcal{L} \to \mathcal{C}$ such that*

$$g(x)'' = g(x) \vee \mathbf{0}'' = h(x) \qquad \textit{for all } x \in L.$$

*Proof.* Suppose the $\Theta$-string $A$ codes $h$. Let $P = \hat{T}(A)$, $g(u_i) = \deg(P^{[i]})$ and $\mathcal{C} = \mathrm{rg}(g)$. Then $g$ is an embedding of $L$ onto an initial segment of $\mathcal{D}$, and $P$ codes it. Furthermore, for all $i \leq n$

$$g(u_i)'' = g(u_i) \vee \mathbf{0}'' = \deg(A^{[i]}) \vee \mathbf{0}'' = \deg(A^{[i]}) \vee h(u_0) = h(u_i).$$

□

Next we show that Theorem 3.3.2 always applies to any finite distributive lattice. We already know ([Ler83, B.1]) that every such lattice has a finite homogeneous lattice table, so it remains to show that any homomorphism from such a lattice is representable with respect to some table. To show this, we follow the construction of the lattice table in [Ler83, B.1], repeated here without the justification. In the following assume that $\mathcal{L}$ is distributive.

**Definition 3.3.3.** $u_i \neq u_0$ is *join-irreducible* if $u_i = u_j \vee u_k$ implies $i = j$ or $i = k$.

Let $\{u_{i_k}\}_{k<r}$ be the join-irreducible elements of $\mathcal{L}$, and let $\mathcal{B} = \mathcal{P}\{a_0, \ldots, a_{r-1}\}$ be the boolean algebra with $r$ atoms (with $\wedge$ and $\vee$ interpreted as set intersection and union, respectively). Define $g : \mathcal{L} \to \mathcal{B}$ by

$$g(u_i) = \{a_k \,:\, u_{i_k} \leq u_i\}$$

**Lemma 3.3.4.** *$g$ is a lattice embedding of $\mathcal{L}$ in $\mathcal{B}$.*

For every $A \subseteq \{a_0, \ldots, a_{r-1}\}$, define a function $\alpha_A : \mathcal{B} \to \omega$ as follows. If $C = \{a_{i_1}, \ldots, a_{i_{|C|}}\}$ with $i_1 < \cdots < i_{|C|}$, then let

$$\alpha_A(C) = \sum\{2^{|C|-j} \,:\, a_{i_j} \in A\}$$

Now, for every $A$ as above and $i \leq n$, let $\alpha^*_A(i) = \alpha_A(g(u_i))$. Let

$$\Theta = \{\alpha^*_A \,:\, A \subseteq \{a_0, \ldots, a_{r-1}\}\}.$$

**Lemma 3.3.5.** *$\Theta$ is a finite homogeneous lattice table for $\mathcal{L}$.*

Now we show that all appropriate homomorphisms from $\mathcal{L}$ can be coded in $\Theta$.

**Lemma 3.3.6.** *Let $\mathcal{L}$ be finite distributive, $\Theta$ as described above, and suppose $h : \mathcal{L} \to \mathcal{D}$ is a u.s.l. homomorphism, with $h(u_0) = \mathbf{0}''$. Then $h$ is $\Theta$-representable.*

*Proof.* Pick sets $A_k \in h(u_{i_k})$ for all $k < r$, and define (for all $x \in \omega$ and $k < r$)

$$A(rx+k) = \begin{cases} \alpha^*_{\{a_k\}} & x \in A_k \\ \alpha^*_\emptyset & x \notin A_k \end{cases}$$

The reason for this choice of lattice elements is that $\alpha^*_{\{a_k\}} \not\equiv_{i_k} \alpha^*_\emptyset$, but $\alpha^*_{\{a_k\}} \equiv_j \alpha^*_\emptyset$ whenever $u_j \not\geq u_{i_k}$.

Fix $k < r$. First note that for all $x \in \omega$

$$x \in A_k \iff A(rx+k) \not\equiv_{i_k} \alpha^*_\emptyset$$

and therefore $A_k \leq_T A^{[i_k]}$. To show the converse, note that for all $j < r$ and $x \in \omega$

$$A^{[i_k]}(rx+j) = \begin{cases} \alpha^*_{\{a_j\}}(i_k) & x \in A_j \\ \alpha^*_\emptyset(i_k) & x \notin A_j \end{cases}$$

If $u_{i_j} \leq u_{i_k}$ then $A_j \leq_T A_k$. Given $A_k$ we can find whether or not $x \in A_j$, then find the required value. If $u_{i_j} \not\leq u_{i_k}$ then $\alpha^*_{\{a_j\}} \equiv_{i_k} \alpha^*_\emptyset$, and consequently $A^{[i_k]}(rx+j) = \alpha^*_\emptyset(i_k)$.

To complete the proof, fix $u_i \in \mathcal{L}$ with $i > 0$. Let $J = \{j : u_{i_j} \leq u_i\}$. Then

$$h(u_i) = \bigvee_{j \in J} h(u_{i_j}) = \bigvee_{j \in J} \deg(A_j) = \bigvee_{j \in J} \deg(A^{[i_j]}) = \deg(A^{[i]}) = \deg(A^{[i]}) \vee h(u_0),$$

and clearly $h(u_0) = \mathbf{0}'' = \deg(A^{[0]}) \vee h(u_0)$. $\square$

The double-jump inversion of finite distributive lattices now follows.

**Theorem 3.3.7.** *Suppose $\mathcal{L}$ is a finite distributive lattice, and $h : \mathcal{L} \to \mathcal{D}$ is an u.s.l. homomorphism, with $h(u_0) = \mathbf{0}''$. Then there is an initial segment $\mathcal{C}$ of $\mathcal{D}$*

*and a lattice isomorphism* $g : \mathcal{L} \to \mathcal{C}$ *such that*

$$g(x)'' = g(x) \vee \mathbf{0}'' = h(x) \qquad \text{for all } x \in L.$$

*Proof.* Let $\Theta$ be as above. Then $h$ is $\Theta$-representable, and the result follows from Theorem 3.3.2. $\square$

Finally, relativization and iteration of the previous two theorems yield the following results, which may be helpful tools in determining the decidability of fragments of $\mathrm{Th}(\mathcal{D}, 0, \leq, \vee, '')$.

**Theorem 3.3.8.** *Suppose* $\mathcal{L}$ *has a finite homogeneous lattice table* $\Theta$, *and* $h : \mathcal{L} \to \mathcal{D}$ *is a* $\Theta$*-representable u.s.l. homomorphism with* $h(u_0) = \mathbf{0}^{(2n)}$. *Then there are initial segments* $\mathcal{C}_i$ *of* $\mathcal{D}(\geq \mathbf{0}^{(2i)})$ *and lattice isomorphisms* $g_i : \mathcal{L} \to \mathcal{C}_i$ *for* $i < n$ *such that*

$$g_i(x)'' = g_i(x) \vee \mathbf{0}^{(2i+2)} = g_{i+1}(x) \qquad \text{for all } x \in L \text{ and } i < n-1, \text{ and}$$

$$g_{n-1}(x)'' = g_{n-1}(x) \vee \mathbf{0}^{(2n)} = h(x) \qquad \text{for all } x \in L.$$

*Proof.* First relativize Theorem 3.3.2 to $\mathbf{0}^{(2n)}$, to get $g_{n-1}$ as required. Notice that $g_{n-1}$ is $\Theta$-representable by the nature of the proof of Theorem 3.3.2. This allows relativization to $\mathbf{0}^{(2n-2)}$. Proceed by induction. $\square$

**Theorem 3.3.9.** *Suppose* $\mathcal{L}_i$ *are finite distributive lattices for* $i \leq n$, *with u.s.l. homomorphisms* $h_i : \mathcal{L}_i \to \mathcal{L}_{i+1}$ *for all* $i < n$. *Suppose further that* $g_n : \mathcal{L}_n \to \mathcal{D}$ *is a u.s.l. isomorphism mapping the bottom element of* $\mathcal{L}_n$ *to* $\mathbf{0}^{(2n)}$. *Then there are initial segments* $\mathcal{C}_i$ *of* $\mathcal{D}(\geq \mathbf{0}^{(2i)})$ *and lattice isomorphisms* $g_i : \mathcal{L}_i \to \mathcal{C}_i$ *for* $i < n$ *such that*

$$g_i(x)'' = g_i(x) \vee \mathbf{0}^{(2i+2)} = g_{i+1}(h_i(x)) \qquad \text{for all } i < n \text{ and } x \in \mathcal{L}_i.$$

*Proof.* This is a fairly straightforward induction on $i$ starting at $n-1$, relativizing Theorem 3.3.7 to $\mathbf{0}^{(2i+2)}$ with $g_{i+1} \circ h_i : \mathcal{L}_i \to \mathcal{D}(\geq \mathbf{0}^{(2i+2)})$ as the u.s.l. homomorphism. Lemma 3.3.6 ensures that the produced embedding $g_i$ is representable with respect to the appropriate lattice table for $\mathcal{L}_i$. □

If we could demonstrate a similar guarantee of representability in the general (non-distributive) case, we could generalize this result. This, however, seems more difficult, if at all possible.

# Chapter 4

# Countable Linear Orders

The narrative in this chapter is similar to the one in the previous chapter. The setting here is borrowed from [Eps79, IV-VI]. Section 2 has the main construction and section 3 has the results.

## 4.1 Introduction

In the following, $\mathcal{L} = (\omega, \leq_{\mathcal{L}})$ is a countable linear order, computable in $\mathbf{0}''$, with bottom element 0 and top element 1. Let $\mathcal{L}_n$ be $\mathcal{L}$ restricted to $\{0, 1, \ldots, n\}$.

As forcing conditions, we use recursive uniform binary trees (i.e. $\{0,1\}$-trees). The linear order will be represented by a collection of recursive sets, ordered by inclusion, and the trees will have densely many branchings which differentiate between distinct sets in the representation. The following definitions make these notions precise.

**Definition 4.1.1.** A binary tree $T$ is *uniform* if for all finite binary strings $\sigma, \tau$

(i) $|\sigma| = |\tau| \leftrightarrow |T(\sigma)| = |T(\tau)|$

(ii) $|\sigma| = |\tau| \rightarrow \forall i < 2 \; T(\sigma \frown i) - T(\sigma) = T(\tau \frown i) - T(\tau)$

**Definition 4.1.2.** Let $X_i \subseteq \omega$ be recursive for $i \in \omega$.

(i) The sequence $X_0, \ldots, X_n$ *represents* $\mathcal{L}_n$ if for all $i, j \leq n$

$$i <_{\mathcal{L}} j \leftrightarrow (X_i \subset^* X_j \wedge |X_j \setminus X_i| = \omega).$$

(ii) The sequence $X_0, X_1, \ldots$ *represents* $\mathcal{L}$ if $X_0, \ldots, X_n$ represents $\mathcal{L}_n$ for all $n \in \omega$.

**Definition 4.1.3.** Recall Definition 2.2.2.

(i) Let $X \subseteq Y \subseteq \omega$. A uniform binary tree $T$ is *special for* $X$ (resp. *for* $-Y$, *for* $X - Y$) if there are infinitely many binary strings $\sigma$ such that $T(\sigma \frown 0)$ and $T(\sigma \frown 1)$ agree on $X$ (resp. disagree on $Y$, agree on $X$ and disagree on $Y$).

(ii) Let $\mathcal{C}_n = \{X_0, \ldots, X_n\}$ represent $\mathcal{L}_n$. A uniform binary tree $T$ is *special for* $\mathcal{C}_n$ if it is special for $X_i - X_j$ for all $i, j \leq n$ such that $i <_{\mathcal{L}} j$. Note that in this case $T \cdot E_\sigma$ is special for $\mathcal{C}_n$ as well, for every $\sigma$.

**Definition 4.1.4.** Let $\mathcal{C} = \{X_0, X_1, \ldots\}$ be a sequence that represents $\mathcal{L}$. An infinite binary string $P$ *codes an embedding of* $\mathcal{L}$ *onto an initial segment of* $\mathcal{D}$ *with respect to* $\mathcal{C}$ if the map $i \mapsto \deg(X_i(P))$ is such an embedding.

Note that the map above is necessarily an order homomorphism by the remark following Definition 2.2.2. In order to make it an embedding onto an initial segment, some requirements need to be forced.

**Definition 4.1.5.** Let $e \in \omega$, $T$ be a binary tree, and $\mathcal{C}_n$ represent $\mathcal{L}_n$.

(i) $T$ *decides* $e$*-totality* if either $\varphi_e^P$ is total for all $P \in [T]$ or $\varphi_e^P$ is not total for all $P \in [T]$. $T$ *forces* $e$*-totality,* $e$*-partiality* in the first, resp. second case.

(ii) $T$ is *$e$-splitting for $\mathcal{C}_n$* if there exists $i \leq n$ such that for all $P \in [T]$ with $\varphi_e^P$ total, we have $\varphi_e^P \equiv_T X_i(P)$.

The definition of $e$-splitting in [Eps79] is different, but implies this one by the Computation Lemma ([Eps79, IV.C]). If the generic path lies on an $e$-splitting tree for all $e$, it will thus code a map onto an initial segment. To make sure that the map does not collapse anywhere, we can force differentiation as we did in Chapter 3. But it is simpler to rely on the general Diagonalization Lemma ([Eps79, IV.C], based on Posner's argument), which we reword as follows.

**Lemma 4.1.6.** *Suppose that the sequence $X_i$ represents $\mathcal{L}$, $i <_{\mathcal{L}} j$, and $B$ is an infinite binary string. Suppose further that the following is true for all but finitely many $n$: that $B$ lies on a uniform binary tree $T$ which is $n$-splitting and special for $X_i - X_j$. Then $X_j(B) \not\leq_T X_i(B)$.*

Finally we state the lemma concerning the effective density of the properties which we would like to force. As in Chapter 3, it is justified by reviewing the arguments in the source ([Eps79, V.A, VI.B]) and noticing that the different cases are defined by two quantifier statements.

**Lemma 4.1.7.** *There is a total function $f : \omega \times \omega \times \omega \to \omega$, computable in $\mathbf{0}''$, such that if $\mathcal{C}_n$ is (a recursive index for) a sequence representing $\mathcal{L}_n$ and $T$ is (a recursive index for) a uniform binary tree, special for $\mathcal{C}_n$ then $f(\mathcal{C}_n, e, T)$ is (a recursive index for) a uniform subtree of $T$, special for $\mathcal{C}_n$ which decides $e$-totality and is $e$-splitting for $\mathcal{C}_n$.*

## 4.2 The Main Construction

We construct a tree of embeddings, as we did in Chapter 3.

**Lemma 4.2.1.** *There is a sequence $\mathcal{C}$ representing $\mathcal{L}$ and a uniform binary tree $\hat{T}$, both computable in $\mathbf{0}''$, such that every $P \in [\hat{T}]$ codes an embedding of $\mathcal{L}$ onto an initial segment of $\mathcal{D}$ with respect to $\mathcal{C}$. Furthermore, there is a recursive sequence $\{Y_i\}$ of disjoint infinite subsets of $\omega$ such that for any path $P = \hat{T}(A)$ and any $i \in \omega$*

$$X_i(P)'' \equiv_T X_i(P) \vee \mathbf{0}'' \equiv_T \oplus_{j \leq_{\mathcal{L}} i} Y_j(A) \vee \mathbf{0}''$$

*Proof.* First we fix the sequence $\{Y_i\}$. Let $l : \omega \to \omega \setminus \{0\}$ be recursive such that

(i) $l(i) \leq i + 1$ for all $i$

(ii) $Y_i = l^{-1}(i)$ is infinite for all $i > 0$

Let $f$ be as in Lemma 4.1.7. We define in stages the sequence $\mathcal{C} = \{X_0, X_1, \dots\}$ and recursive uniform trees $T_\sigma$ for all finite binary strings $\sigma$. We use $\mathcal{C}_n$ to denote the partial sequence $X_0, \dots, X_n$. The trees $T_\sigma$ will satisfy the following properties:

(i) $T_\sigma$ is special for $\mathcal{C}_{|\sigma|+1}$

(ii) If $|\sigma| = j$ then $|T_\sigma(\emptyset)| \leq \min(X_{j+1})$

(iii) If $|\sigma| = j$ then $T_\sigma(0)$ and $T_\sigma(1)$ disagree on $X_{l(j)}$, and $l(j)$ is the $\mathcal{L}_{j+1}$-least for which this holds.

(iii)* If $|\sigma| = j$ then $T_{\sigma\frown 0}(\emptyset)$ and $T_{\sigma\frown 1}(\emptyset)$ disagree on $X_{l(j)}$, and $l(j)$ is the $\mathcal{L}_{j+1}$-least for which this holds.

(iv) $T_{\sigma\frown d} \subseteq T_\sigma E_d$ (for $d = 0, 1$)

(v) $T_{\sigma\frown d} \subseteq f(\mathcal{C}_{|\sigma|+1}, |\sigma|, T)$ for some recursive tree $T$

(vi) if $|\sigma_1| = |\sigma_2|$ then $|T_{\sigma_1}(\emptyset)| = |T_{\sigma_2}(\emptyset)|$

(vii) if $|\sigma_1| = |\sigma_2|$ then

$$T_{\sigma_1}(\rho \frown \rho') - T_{\sigma_1}(\rho) = T_{\sigma_2}(\rho \frown \rho') - T_{\sigma_2}(\rho).$$

Let $X_0 = \emptyset$ and $X_1 = \omega$. We let $T_\emptyset = T_{\mathrm{id}}$, and note that it is trivially special for $\mathcal{C}_1$ since *any* branching agrees on $X_0$ but not on $X_1$. Property (iii) is also fulfilled, since $l(0) = 1$. Suppose $T_\sigma$ has been defined for all $\sigma$ of length $k$. Let $\{\tau_j\}_{j=1}^m$ list all binary strings of length $k+1$, and let $\tau_j = \sigma_j \frown d_j$ with $|\sigma_j| = k$ and $d_j < 2$.Let $T'_{\tau_j} = T_{\sigma_j} E_{d_j}$. We go by subinduction on $j$:

$$T^*_{\tau_1} = f(\mathcal{C}_{k+1}, k, T'_{\tau_1}), \qquad T^*_{\tau_{j+1}} = f(\mathcal{C}_{k+1}, k, T'_{\tau_{j+1}}(T^*_{\tau_j}/T'_{\tau_j})).$$

For all $1 \leq j \leq m$ now let

$$\tilde{T}_{\tau_j} = T'_{\tau_j}(T^*_{\tau_m}/T'_{\tau_m}).$$

As in Chapter 3, $\tilde{T}_{\tau_j}$ are uniform recursive and satisfy conditions (iv)-(vii) above (in place of $T_{\tau_j}$. Furthermore, it is easily seen that replacing these by $\tilde{T}_{\tau_j} E_\sigma$ for any fixed $\sigma$ will maintain that.

In order to satisfy property (i), we simply define $X_{k+2}$ in a way which would make the trees $\tilde{T}_{\tau_j}$ special for $\mathcal{C}_{k+2}$. Let $a$ and $b$ be the immediate predecessor and immediate successor (resp.) of $k+2$ in $\mathcal{L}_{k+2}$. Now, $\tilde{T}_{\tau_1}$ is special for $\mathcal{C}_{k+1}$, and in particular for $X_a - X_b$, so that the set

$$Z = \{m : |\sigma| = m \to \tilde{T}_{\tau_1}(\sigma \frown 0), \tilde{T}_{\tau_1}(\sigma \frown 1) \text{ agree on } X_a \text{ but not on } X_b\}$$

is infinite (and clearly recursive). Let $m_0, m_1, \ldots$ be an increasing enumeration of $Z$. Let

$$Z_0 = \{x : x \in X_b \setminus X_a \text{ and } |\tilde{T}_{\tau_1}(\sigma)| \leq x < |\tilde{T}_{\tau_1}(\sigma \frown 0)| \text{ where } |\sigma| = m_{2t},\, t \in \omega\},$$

an infinite recursive set. Let $\tilde{X}_{k+2} = X_a \cup Z_0$. Then $X_a \subseteq^* \tilde{X}_{k+2} \subseteq^* X_b$ and the differences are infinite. Also, $\tilde{T}_{\tau_1}$ is special for $X_a - \tilde{X}_{k+2}$ since at any level $m_{2t}$ it branches in a way which agrees on $X_a$ and disagrees on $\tilde{X}_{k+2}$. By a similar argument (for levels $m_{2t+1}$) it is seen to be special for $\tilde{X}_{k+2} - X_b$. Now note that for any $1 \leq j \leq m$ and any $\sigma$, the tree $\tilde{T}_{\tau_j} E_\sigma$ will be special for $X_a - \tilde{X}_{k+2}$ and for $\tilde{X}_{k+2} - X_b$, because the branchings in all trees $\tilde{T}_{\tau_j}$ are the same (property (vii)) and taking an extension tree will only remove finitely many levels.

For condition (iii), let $b = l(k+1)$ and $a$ the immediate predecessor of $b$ in $\mathcal{L}_{k+2}$. Let $r$ be the least such that $\tilde{T}_{\tau_1}(0^r 0)$ and $\tilde{T}_{\tau_1}(0^r 1)$ agree on $X_a$ and not on $X_b$. Such $r$ exists, since the tree is special for $X_a - X_b$. Now let $T_{\tau_j} = \tilde{T}_{\tau_j} E_{0^r}$ for all $1 \leq j \leq m$. This takes care of condition (iii). Condition (iii)$^*$ is fulfilled as well, since for any $d < 2$

$$T_{\sigma_j \frown d}(\emptyset) = \tilde{T}_{\sigma_j \frown d}(0^r) = T'_{\sigma_j \frown d}(T^*_{\tau_m}/T'_{\tau_m})(0^r) = T_{\sigma_j} E_d(\rho) = T_{\sigma_j}(d \frown \rho)$$

where $\rho = (T^*_{\tau_m}/T'_{\tau_m})(0^r)$. By condition (vii), $T_{\sigma_j}(d \frown \rho) - T_{\sigma_j}(d)$ is the same for both values of $d$, and thus the disagreement between the strings $T_{\sigma_j \frown d}(\emptyset)$ is the same as between the strings $T_{\sigma_j}(d)$. Condition (iii)$^*$ now follows immediately from condition (iii).

Finally, let $X_{k+2} = \tilde{X}_{k+2} \setminus \{0, 1, \ldots, |T_{0^{k+1}}(\emptyset)|\}$. As this is only a finite change, we still have $X_a \subseteq^* X_{k+2} \subseteq^* X_b$ with infinite differences, and $T_{\tau_j}$ is special for $X_a - X_{k+2}$ and for $X_{k+2} - X_b$, for all $1 \leq j \leq m$. Therefore $\mathcal{C}_{k+2}$ represents $\mathcal{L}_{k+2}$, and the trees $T_{\tau_j}$ are special for $\mathcal{C}_{k+2}$. Condition (ii) is now fulfilled by definition of $X_{k+2}$. This completes the induction on $k$.

Now let $\hat{T}(\sigma) = T_\sigma(\emptyset)$. By an argument analogous to that in Chapter 3, $\hat{T}$ is uniform. The tree, as well as the sequence $\mathcal{C} = \cup\mathcal{C}_k$ which represents $\mathcal{L}$, are recursive in $\mathbf{0}''$ since the construction is.

Fix $P = \hat{T}(A) \in [\hat{T}]$. The requirements forced, together with lemma 4.1.6, imply that $P$ codes an embedding of $\mathcal{L}$ onto an initial segment of $\mathcal{D}$. It is left to show that

$$X_i(P) \vee \mathbf{0}'' \geq_T \oplus_{j \leq_{\mathcal{L}} i} Y_j(A) \vee \mathbf{0}'' \geq_T X_i(P)''$$

holds for every $i \in \omega$.

To establish the first inequality, it suffices to demonstrate that $X_i(P) \vee \mathbf{0}'' \geq_T Y_i(A)$ uniformly for all $i$, since then for all $j \leq_{\mathcal{L}} i$ we have

$$X_i(P) \vee \mathbf{0}'' \geq_T X_j(P) \vee \mathbf{0}'' \geq_T Y_j(A).$$

(The reduction on the left is uniform because the sequence $\{X_i\}$ is uniformly recursive in $\mathbf{0}''$.) We compute the value of $Y_i(A)(n)$ given $X_i(P) \vee \mathbf{0}''$. First let $m$ be the $n$-th value of $Y_i$, so that $Y_i(A)(n) = A(m)$. Note that $l(m) = i$, and therefore $T_{\sigma\frown 0}(\emptyset)$ and $T_{\sigma\frown 1}(\emptyset)$ disagree on $X_i$ whenever $|\sigma| = m$, by property (iii)*. Then $A(m) = d$ such that $T(\sigma \frown d)$ agrees with $P$ on $X_i$ above $|T(\sigma)|$.

Next we show that $\oplus_{j \leq_{\mathcal{L}} i} Y_j(A) \vee \mathbf{0}'' \geq_T X_i(P)''$. To find $X_i(P)''(n)$, we need to find whether $\varphi_n^{X_i(P)}$ is total or not. First let $m > 0$ be such that $\varphi_n^{X_i(P)} \simeq \varphi_m^P$ and $\varphi_m$ depends only on oracle values at members of $X_i$ ($\varphi_m$ first applies $X_i$ to the oracle, then computes $\varphi_n$ with the new oracle). We find, by induction on $j \leq m$, a string $\sigma$ of length $j + 1$ such that $T_\sigma(\emptyset)$ agrees with $P$ on $X_i$. Suppose that we have $\sigma$ up to length $j$. If $T_{0^j\frown 0}(\emptyset)$ and $T_{0^j\frown 1}(\emptyset)$ agree on $X_i$, let $\sigma(j) = 0$ (a value of 1 would do just as well, of course). Otherwise, $T_{0^j\frown 0}(\emptyset)$ and $T_{0^j\frown 1}(\emptyset)$ disagree on $X_i$. We cannot have $i > j + 1$, since then by condition (ii)

$$|T_{0^{j+1}}| \leq |T_{0^{i-1}}(\emptyset)| \leq \min(X_i),$$

in contradiction to the disagreement. Thus $i \in \mathcal{L}_{j+1}$, and by condition (iii)* we have $l(j) \leq_{\mathcal{L}} i$ (since $l(j)$ is the least in $\mathcal{L}_{j+1}$ for which there is a disagreement). $Y_{l(j)}(A)$ is then part of our oracle, and we can find $A(j) = Y_{l(j)}(A)(k)$ (the value of $k$ depends on $l$ alone). Set $\sigma(j) = A(j)$. This must be the branching which agrees with $P$ on $X_i$, since the tree is uniform. This completes the computation of $\sigma$.

Now notice that

$$\begin{aligned} n \in X_i(P)'' &\iff \varphi_n^{X_i(P)} \text{ is total} \\ &\iff \varphi_m^P \text{ is total} \\ &\iff T_{A|(m+1)} \text{ forces } m\text{-totality} \\ &\iff T_\sigma \text{ forces } m\text{-totality}, \end{aligned}$$

where the last equivalence stems from the equality

$$\{X_i(Q) : Q \in [T_{A|(m+1)}]\} = \{X_i(Q) : Q \in [T_\sigma]\}$$

which is a consequence of property (vii) and the agreement of $T_\sigma(\emptyset)$ and $P = \hat{T}(A)$ on $X_i$. $\square$

## 4.3 The Results

As in Chapter 3, we first conclude

**Corollary 4.3.1.** *There are $2^\omega$ many lattice embeddings of $\mathcal{L}$ as an initial segment of $\mathcal{D}$.*

On to the main theorem of this section.

**Theorem 4.3.2.** *Let $\mathcal{L} = (\omega, \leq)$ be a countable linear order, computable in $\mathbf{0}''$, with bottom element 0 and top element 1. Let $h : \mathcal{L} \to \mathcal{D}$ be an order homomorphism, with $h(0) = \mathbf{0}''$. There is an initial segment $\mathcal{C}$ of $\mathcal{D}$ and an order isomorphism $g : \mathcal{L} \to \mathcal{C}$ such that*

$$g(i)'' = g(i) \vee \mathbf{0}'' = h(i) \qquad \textit{for all } i \in L.$$

*Proof.* In view of Lemma 4.2.1, all we need is to define $A$ in such a way that $\deg(\oplus_{j \leq_{\mathcal{L}} i} Y_j(A)) \vee \mathbf{0}'' = h(i)$ for all $i \in w$. Pick representatives $A_i \in h(i)$ for $i > 0$, and let $A$ be such that $Y_i(A) = A_i$ for all $i \in \omega$ (such $A$ is unique, since the sets $Y_i$ form a partition of $\omega$). Now observe that for all $i > 0$

$$h(i) = \deg(A_i) = \deg(Y_i(A)) = \deg(\oplus_{j \leq_{\mathcal{L}} i} Y_j(A)) \vee \mathbf{0}''$$

since $h(i) \geq \mathbf{0}''$ for all $i > 0$ and $A_i \geq_T A_j$ uniformly for $j \leq_{\mathcal{L}} i$ (the uniformity follows from the fact that the sequence $\{Y_i\}$ is uniformly recursive.

Now let $P = \hat{T}(A)$, and set $g(i) = \deg(X_i(P))$. The equality in the theorem follows for $i > 0$ by the above, and for $i = 0$ trivially. $\square$

We can relativize and iterate to obtain

**Theorem 4.3.3.** *Let* $\mathcal{L} = (\omega, \leq)$ *be a countable linear order, computable in* $\mathbf{0}''$*, with bottom element* $0$ *and top element* $1$*. Let* $h : \mathcal{L} \to \mathcal{D}$ *be an order homomorphism, with* $h(0) = \mathbf{0}^{(2n)}$*. There are initial segments* $\mathcal{C}_k$ *of* $\mathcal{D}(\geq \mathbf{0}^{(2k)})$ *and order isomorphisms* $g_k : \mathcal{L} \to \mathcal{C}_k$ *for* $k < n$ *such that*

$$
\begin{aligned}
g_k(i)'' &= g_k(i) \vee \mathbf{0}^{(2k+2)} = g_{k+1}(i) && \text{for all } i \in \omega \text{ and } k < n-1, \text{ and} \\
g_{n-1}(i)'' &= g_{n-1}(i) \vee \mathbf{0}^{(2n)} = h(i) && \text{for all } i \in \omega.
\end{aligned}
$$

*Proof.* This is straightforward. Since $\mathcal{L}$ is recursive in $\mathbf{0}''$, it is recursive in $\mathbf{0}^{(2k)}$ for any $k > 0$. $\square$

# Chapter 5

# Degrees r.e.a. in $\mathbf{0}''$

Section 1 deduces the main result from known theorems. Section 2 presents a direct construction which produces a slight alteration of it. Section 3 explains why a generalization of the result to non-trivial lattices is problematic.

## 5.1 Introduction

The following theorem is a relativization of the Sacks Inversion Theorem. The second one is the Cooper Inversion Theorem.

**Theorem 5.1.1.** [Sac63] *If* $\mathbf{b}$ *is r.e. in and above* $\mathbf{x}'$ *then there exists a degree* $\mathbf{c}$ *r.e. in and above* $\mathbf{x}$ *such that* $\mathbf{c}' = \mathbf{b}$*.*

**Theorem 5.1.2.** [Coo73] *If* $\mathbf{c} \geq \mathbf{0}'$ *then there exists a minimal degree* $\mathbf{a}$ *such that* $\mathbf{a}' = \mathbf{c}$.

If we take $\mathbf{x}$ to be $\mathbf{0}'$ in Theorem 5.1.1, the two theorems combine to produce

**Theorem 5.1.3.** *If* $\mathbf{b}$ *is r.e. in and above* $\mathbf{0}''$ *then there exists a minimal degree* $\mathbf{a} < \mathbf{0}''$ *such that* $\mathbf{b} = \mathbf{a}''$*.*

*Proof.* We get $\mathbf{a} < \mathbf{0}''$ since $\mathbf{c} = \mathbf{a}'$ is r.e. in $\mathbf{0}'$, and so computable in $\mathbf{0}''$. $\square$

Note that the converse also holds: if $\mathbf{a} < \mathbf{0}''$ is minimal, then $\mathbf{a}''$ is r.e. in (and trivially above) $\mathbf{0}''$, for $\mathbf{a}'' = (\mathbf{a} \vee \mathbf{0}')'$ by Theorem 2.1.5, and that is r.e. in $\mathbf{a} \vee \mathbf{0}' \leq \mathbf{0}''$. This characterizes the degrees r.e. in and above $\mathbf{0}''$ as the double jumps of minimal degrees below $\mathbf{0}''$. As mentioned in Chapter 1, the corresponding problem for degrees r.e. in and above $\mathbf{0}'$ is not yet solved, and any solution will not be as simple as in the case of $\mathbf{0}''$.

In the following section we directly construct a double jump inversion of a degree r.e. in and above $\mathbf{0}''$, in a manner which is considerably simpler than the sum of the proofs of the two theorems above. We build on the standard construction of a minimal through forcing with binary trees (uniformity is not needed here), and we code a set $B$ of degree $\mathbf{b}$ as we go along. However, we only have an enumeration of $B$ (since the oracle is $\mathbf{0}''$), and as a result we may make wrong coding steps. Going back to fix these mistakes creates finite injury, and the conditions settle down eventually.

It turns out that the degree $\mathbf{a}$ produced by this construction is different in a fundamental way from the degree given by the proof of Theorem 5.1.3. For we can reconstruct the set $B$ using $\mathbf{a}'$ and the construction oracle, giving

$$\mathbf{b} = \mathbf{a}' \vee \mathbf{0}'',$$

whereas in Theorem 5.1.3 we have $\mathbf{a}' \leq \mathbf{0}''$, and so $\mathbf{a}' \vee \mathbf{0}'' = \mathbf{0}''$.

Before presenting the construction, we mention the appropriate density lemma, justified in [Ler83, V.2.7, V.3.2].

**Definition 5.1.4.** A binary tree $T$ is *e-splitting* if for all $P \in [T]$, whenever $\varphi_e^P$ is total it is either recursive or computes $P$.

**Lemma 5.1.5.** *There is a total function $f : \omega \times \omega \to \omega$, computable in $\mathbf{0}''$, such that if $T$ is (a recursive index for) a binary tree then $f(e, T)$ is (a recursive index for) a binary tree which*

*(i) is a subtree of $T \cdot E_i$ where $i < 2$ is such that $T(i)$ is incompatible with $\varphi_e$;*

*(ii) decides e-totality;*

*(iii) is e-splitting.*

We also define the *standard narrow binary tree* $N_2$ as follows:

$$N_2(d_0 d_1 \dots d_n) = d_0 0 d_1 0 \dots d_n 0$$

for all $n \in \omega$, $d_i < 2$. Note that for every binary tree $T$, the tree $T \cdot N_2$ is a narrow subtree of $T$: for no binary string $\sigma$ is $T \cdot E_\sigma$ a subtree of $T \cdot N_2$. Moreover, if $T$ is recursive, so is $T \cdot N_2$.

## 5.2 The Construction

**Theorem 5.2.1.** *If* $\mathbf{b}$ *is r.e. in and above* $\mathbf{0}''$ *then there exists a minimal degree* $\mathbf{a} < \mathbf{0}''$ *such that*

$$\mathbf{b} = \mathbf{a}'' = \mathbf{a}' \vee \mathbf{0}'' = (\mathbf{a} \vee \mathbf{0}')'.$$

*Proof.* Fix a set $B \in b$ r.e. in $\mathbf{0}''$, and let $\langle b_i \rangle_{i \in \omega}$ be an enumeration witnessing that. Let $B_s = \{b_i : i < s\}$ approximate $B$. The construction is a finite injury one, with oracle $\mathbf{0}''$. At stage $s$ we define a nested sequence of recursive binary trees $T_i^s$ for $i \leq g(s)$, and a finite string $\alpha_s = T^s_{g(s)}(\emptyset)$.

Let $g(0) = 0$, and let $T_0^0$ be the identity tree (with $\alpha_0 = \emptyset$). At stage $s+1$, if $g(s) > b_s$ (injury) let $g(s+1) = b_s + 1$, otherwise let $g(s+1) = g(s) + 1$. For all $i < g(s+1)$ let $T_i^{s+1} = T_i^s$. Let $n = g(s+1) - 1$. If $n \notin B_{s+1}$ let $T_{g(s+1)}^{s+1} = f(n, T_n^{s+1} N_2)$. Otherwise let $T_{g(s+1)}^{s+1} = f(n, T_n^{s+1} E_\sigma)$ where $\sigma$ is the first string of even length $m$ such that $\sigma(m-1) = 1$ and $T_n^{s+1}(\sigma) \supseteq \alpha_s$. In any case, let $\alpha_{s+1} = T_{g(s+1)}^{s+1}(\emptyset)$.

Let $A = \cup_s \alpha_s$, $\mathbf{a} = \deg(A)$. This completes the construction.

Before we proceed with the proof, note that in case of injury ($g(s) > b_s$), $T_{b_s+1}^{s+1}$ is defined to be a subtree of $T_{b_s}^s E_\sigma$ where $\sigma$ is not on $N_2$, whereas at stage $s$, $T_{b_s+1}^s \subseteq T_{b_s} N_2$. Consequently, not only are $T_{b_s+1}^s$ and $T_{b_s+1}^{s+1}$ different, but they also have disjoint sets of paths. The importance of this point will be apparent in the proof.

First we show by induction that $T_i = \lim_s T_i^s$ exists for all $i$. $T_0^s$ is never changed, since $g(s) > 0$ for all $s > 0$. Let $s$ be a stage at which $T_j^s = T_j$ for all $j \leq i$. This implies that no $n < i$ enters $B$ after $s$. Let $t \geq s$ be the least such that $T_{i+1}^t \neq T_{i+1}^{t+1}$ (if there is no such, we're done). It must be that $b_t = i$, and so $g(t+1) = i+1$. Therefore after $t+1$ no $n \leq i$ enters $B$, $g$ remains above $i+1$, and $\lim_s T_{i+1}^s = T_{i+1}^{t+1}$.

Next, observe that $\mathbf{a}$ is a minimal degree, since for every $e$ there is some tree $S$ such that $A$ lies on $T_{e+1} = f(e, S)$. The construction is recursive in $\mathbf{0}''$, consequently $\mathbf{a} < \mathbf{0}''$.

To find $\mathrm{Tot}^A(n)$ given $B$, follow the construction (using $\mathbf{0}'' \leq B$) until $B_s \restriction (n+1) = B \restriction (n+1)$. As above, this means that $T_{n+1} = T_{n+1}^s = f(n, S)$ for some $S$. Since $A$ is a path on $T_{n+1}$, we have $n \in \mathrm{Tot}^A$ iff $T_{n+1}^s$ forces $n$-totality, and this can be found using $\mathbf{0}''$. Hence $\mathbf{a}'' \leq \mathbf{b}$.

To find $B(n)$ given $A' \vee \mathbf{0}''$, follow the construction, and at every stage $s$ where $T^s_{n+1}$ is redefined, find if $A$ lies on $T^s_{n+1}$ (this is recursive in $A'$). Since $A \in [T_{n+1}]$, this will eventually be the case. We claim that $n \in B$ iff $n \in B_s$, where $s$ is the first such that $A \in [T^s_{n+1}]$. Indeed, if $n \in B_s$, clearly $n \in B$. Suppose $n \notin B_s$. Let $m = b_t$ be the least with $t \geq s$, and suppose that $m \leq n$. Then

$$A \in [T^s_{n+1}] \subseteq [T^s_{m+1}] = [T^t_{m+1}] \quad \text{and} \quad [T^t_{m+1}] \cap [T^{t+1}_{m+1}] = \emptyset,$$

hence $A \notin [T^{t+1}_{m+1}]$. But $T_{m+1} = T^{t+1}_{m+1}$ and $A$ does lie on $T_{m+1}$. The contradiction implies that $m > n$, and consequently that $n$ never enters $B$.

Together with the inequality $\mathbf{a}' \vee \mathbf{0}'' \leq \mathbf{a}''$, this proves the first two equalities in the theorem. The last one follows from Lemma 2.1.5. $\square$

## 5.3 Limitations

It is natural to hope for a generalization of the construction which would allow double-jump inversions of certain lattices (of degrees r.e.a. in $\mathbf{0}''$) onto initial segments. As it turns out, even Theorem 5.1.3 is not generalizable to the simplest lattices.

**Theorem 5.3.1.** *There are degrees $\mathbf{b}_1$ and $\mathbf{b}_2$, r.e. in and above $\mathbf{0}''$, such that for no four element initial segment $\{\mathbf{0}, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_1 \vee \mathbf{a}_2\}$ below $\mathbf{0}''$ do we have $\mathbf{a}_i'' = \mathbf{b}_i$ and $(\mathbf{a}_1 \vee \mathbf{a}_2)'' = \mathbf{b}_1 \vee \mathbf{b}_2$.*

*Proof.* We relativize Shore's non-inversion theorem [Sho88, 1.2] to $\mathbf{0}'$: there are degrees $\mathbf{b}_1$ and $\mathbf{b}_2$, r.e. in and above $\mathbf{0}''$, such that for no $\mathbf{0}' \leq \mathbf{c}_1, \mathbf{c}_2 \leq \mathbf{0}''$ do we have $\mathbf{c}_i' = \mathbf{b}_i$ and $(\mathbf{c}_1 \vee \mathbf{c}_2)' = \mathbf{b}_1 \vee \mathbf{b}_2$. Fix such $\mathbf{b}_1$ and $\mathbf{b}_2$.

Suppose there is an initial segment as in the theorem, and let $\mathbf{c}_i = \mathbf{a}_i \vee \mathbf{0}'$ for $i = 1, 2$. Then $\mathbf{0}' \leq \mathbf{c}_1, \mathbf{c}_2 \leq \mathbf{0}''$, and for $i = 1, 2$ we have

$$\mathbf{c}_i' = (\mathbf{a}_i \vee \mathbf{0}')' = \mathbf{a}_i'' = \mathbf{b}_i$$

since $\mathbf{a}_i$ is $\mathbf{GL}_2$ by Lemma 2.1.5. For a contradiction, note that $(\mathbf{a}_1 \vee \mathbf{a}_2)$ is $\mathbf{GL}_2$ as well, and therefore

$$(\mathbf{c}_1 \vee \mathbf{c}_2)' = ((\mathbf{a}_1 \vee \mathbf{a}_2) \vee \mathbf{0}')' = (\mathbf{a}_1 \vee \mathbf{a}_2)'' = \mathbf{b}_1 \vee \mathbf{b}_2.$$

$\square$

# Chapter 6

# A Non-Weakly Recursive Minimal With a Strong Minimal Cover

We enter the second part of the thesis, concerning weakly recursive degrees and strong minimal covers. The introduction to this section covers the basic definitions and gives an overview of the construction, which follows in section 2.

## 6.1 Introduction

The following definition applies to any partial order. However we restrict our attention to Turing degrees.

**Definition 6.1.1.** Let $\mathbf{b} < \mathbf{a}$ be Turing degrees.

(i) $\mathbf{a}$ is a *minimal cover* of $\mathbf{b}$ if the interval $(\mathbf{b}, \mathbf{a})$ is empty.

(ii) $\mathbf{a}$ is a *strong minimal cover* of $\mathbf{b}$ if the intervals $[\mathbf{0}, \mathbf{b}]$ and $[\mathbf{0}, \mathbf{a})$ are equal.

Clearly (ii) implies (i). The following definition and theorem are due to Ishmukhametov.

**Definition 6.1.2.** A degree $\mathbf{a}$ is *weakly recursive* if there is a recursive function $p$ such that for every function $f \leq \mathbf{a}$ there is a recursive function $h$ such that $|W_{h(n)}| \leq p(n)$ and $f(n) \in W_{h(n)}$ for all $n \in \omega$.

**Theorem 6.1.3.** [Ish99] *Every weakly recursive degree has a strong minimal cover.*

While the converse is true for r.e. degrees [Ish99], we show that it is not true in general. Indeed, in this chapter we construct a (minimal) degree $\mathbf{b}$ which is not weakly recursive but has a strong minimal cover. The $\mathbf{0}''$ construction is based on that of a three element linear initial segment [Eps79, III] with uniform trees. In order to make the middle degree non-weakly recursive, we define functions $f_i \leq \mathbf{b}$ and diagonalize with $f_i$ against $p_i, h_j$ for all $j$, where $p_k = h_k$ is an enumeration of all recursive functions (which our oracle provides).

The diagonalization requires fattening the trees so that the branching exceeds the bounds given by $p_i$. This leads us to abandon binary trees in favor of partial $\omega$-trees. The trees are finitely branching, but all of their paths are infinite (so that their partiality is still limited).

Since the trees diverge from the definition given in Chapter 2, care must be taken when dealing with concepts such as tree division (which is used, but in a limited capacity). Also, we are forced to go into the semi-gory details of splitting trees (which were elegantly skipped in previous chapters).

Recall Definition 2.2.2, and let Od be the set of odd natural numbers.

**Definition 6.1.4.** (i) Two strings $\sigma, \tau$ form an *e-split* if $\varphi_e^\sigma(x)\downarrow \neq \varphi_e^\tau(x)\downarrow$ for some $x \in \omega$.

(ii) A tree is *e-non-splitting* if no pair of strings in its range form an $e$-split.

(iii) A tree is *e-splitting* if every pair of non-compatible strings in its range form an $e$-split.

(iv) A tree $T$ is *e-splitting on the odds* if every branching pair on $T$ (i.e. two immediate successors of the same node) which disagrees on Od forms an $e$-split, and no pair of strings on $T$ which agrees on the odds forms an $e$-split.

The following is a summary of the Computation Lemmas from [Eps79, I.G, III.A].

**Lemma 6.1.5.** *Let $A$ be a path on a tree $T$, such that $C = \varphi_e^A$ is a total function.*

*(i) If $T$ is e-non-splitting, then $C$ is recursive.*

*(ii) If $T$ is e-splitting, then $C \geq_T A$.*

*(iii) If $T$ is e-splitting on the odds, then $C \equiv_T \mathrm{Od}(A)$.*

## 6.2 The Construction

**Theorem 6.2.1.** *There are degrees $\mathbf{b} < \mathbf{a} \leq \mathbf{0}''$ such that $\mathbf{b}$ is minimal and not weakly recursive, and $\mathbf{a}$ is a strong minimal cover of $\mathbf{b}$.*

*Proof.* We construct, with oracle $\mathbf{0}''$, an infinite $\omega$-string $A$ and functions $\{f_i\}_{i\in\omega}$ recursive in $\mathrm{Od}(A)$ satisfying the following requirements:

$$R_s^0: \qquad \mathrm{Od}(A) \neq \varphi_s$$

$$R_s^1: \qquad A \neq \varphi_s^{\mathrm{Od}(A)}$$

$$R_s^2: \qquad s = \langle i, j\rangle \rightarrow \exists n \ \left(\left|W_{h_j(n)}\right| > p_i(n) \ \vee \ f_i(n) \notin W_{h_j(n)}\right)$$

$$R_s^3: \qquad \varphi_s^A = C \text{ is total} \rightarrow (C \leq_T \mathbf{0} \ \vee \ A \leq_T C \ \vee \ C \equiv_T \mathrm{Od}(A))$$

Where $p_i = h_i$ is the $i$-th recursive function (which can be found recursively in $\mathbf{0}''$). Clearly, if $A$ satisfies these requirements for all $k \in \omega$, then $\mathbf{b} = \deg(\mathrm{Od}(A))$ is minimal and not weakly recursive, and $\mathbf{a} = \deg(A)$ is a strong minimal cover of $\mathbf{b}$. The construction will produce a sequence of recursive uniform partial $\omega$-trees such that $T_{n+1} \subseteq T_n$ for all $n$. We use $\tau \sqsubset_i \tau'$ to mean that for some string $\sigma$ and $k \in \omega$, $\tau = T_i(\sigma)$ and $\tau' = T_i(\sigma \frown k)$ (in other words, $\tau'$ is an immediate successor of $\tau$ on $T_i$). We want the following conditions to hold for all $i$:

(i) If $\sigma \frown k \in \mathrm{dom}(T_i)$ and $k' < k$ then $\sigma \frown k' \in \mathrm{dom}(T_i)$.

(ii) $T_i$ is 2-branching at odd levels, i.e. if $\sigma \in \mathrm{dom}(T_i)$ is of odd length then $T_i(\sigma \frown 0)$ and $T_i(\sigma \frown 1)$ are the (only) immediate successors of $T_i(\sigma)$ on $T_i$.

(iii) $(\tau_1, \tau_2 \sqsupset_i T_i(\sigma) \wedge \tau_1 \neq \tau_2) \;\rightarrow\; (|\sigma| \text{ is odd} \leftrightarrow \mathrm{Od}(\tau_1) = \mathrm{Od}(\tau_2))$

(iv) $(\tau_1 \in \mathrm{rg}(T_i) \;\wedge\; i' < i \;\wedge\; \tau_1 \sqsubset_{i'} \tau_2) \rightarrow \exists \tau_3 \supseteq \tau_2 \; (\tau_1 \sqsubset_i \tau_3)$

Note that if $T_i$ satisfies these conditions and $T_{i+1}$ is an extension subtree $T_i \cdot E_\sigma$ with $|\sigma|$ even, then $T_{i+1}$ satisfies these conditions as well.

$A$ will be the unique path which lies on all $T_i$, and the functions $f_i$ are computed from $A$ as follows:

$$f_i(n) = C(2n) \quad \text{where } A = T_{5i}(C).$$

Thus, in order to force the value of $f_i(n)$ it suffices to specify which node on level $2n+1$ of $T_{5i}$ is an initial segment of $A$. Note that by condition (iii), if $C_1(2n) \neq C_2(2n)$ then $\mathrm{Od}(T_{5i}(C_1)) \neq \mathrm{Od}(T_{5i}(C_2))$, and therefore $f_i$ is computable in $\mathrm{Od}(A)$ (for all $i$).

To define $T_0$, let

$$T_0(k_0 d_0 k_1 d_1 k_2 d_2 \ldots) = (0k_0) \frown (d_0 0) \frown (0k_1) \frown (d_1 0) \frown (0k_2) \frown (d_2 0) \frown \ldots$$

for all $k_n \leq \max(p_0(n), 1)$ and $d_n = 0, 1$ (the parenthesis are for readability). Note that $T_0$ satisfies conditions (i)-(iii), and has $> p_0(n)$-branching at level $2n$ (for all $n$). Stages $5s+4$ of the construction will ensure that $T_{5s}$ will be $> p_s(n)$-branching at level $2n$, for all $s > 0$ as well (this will be assumed in stages $5s+2$).

**Stage** $5s$**:** Pick the first $\sigma$ of length 2 such that $\mathrm{Od}(T_{5s}(\sigma))$ is incompatible with $\varphi_s$ (this is recursive in $\mathbf{0}'$). Let $T_{5s+1} = T_{5s}E_\sigma$. Then $T_{5s+1}$ forces $R^0_s$, and conditions (i)-(iv) are maintained as in the remark above.

**Stage** $5s+1$**:** Let $\alpha_i = T_{5s+1}(0 \frown i)$ for $i = 0, 1$, and let $x$ be the first on which $\alpha_0, \alpha_1$ disagree (note that $\mathrm{Od}(\alpha_0) = \mathrm{Od}(\alpha_1)$, so $x$ is even). If there is an extension of $\alpha_0$ on $T_{5s+1}$ which forces $\varphi_s^{\mathrm{Od}(A)}(x)\uparrow$, let $\sigma$ to be the first of even length such that $T_{5s+1}(\sigma)$ is such an extension (this is recursive in $\mathbf{0}''$).

Otherwise let $\sigma_0$ be the first of even length such that $\beta_0 = T_{5s+1}(\sigma_0) = \alpha_0 \frown \tau$ satisfies $\varphi_s^{\mathrm{Od}(\beta_0)}(x)\downarrow$. Let $\beta_1 = \alpha_1 \frown \tau = T_{5s+1}(\sigma_1)$ (exists by uniformity). Then

$$\beta_0(x) \neq \beta_1(x) \quad \text{and} \quad \varphi_s^{\mathrm{Od}(\beta_0)}(x)\downarrow = \varphi_s^{\mathrm{Od}(\beta_1)}(x)\downarrow$$

since $\mathrm{Od}(\beta_0) = \mathrm{Od}(\beta_1)$. Let $i$ be the first such that $\varphi_s^{\mathrm{Od}(\beta_i)}(x) \neq \beta_i(x)$, and let $\sigma = \sigma_i$.

In any case, now let $T_{5s+2} = T_{5s+1}E_\sigma$. Then $T_{5s+2}$ is recursive and forces $R^1_s$, and conditions (i)-(iv) are maintained.

**Stage** $5s+2$**:** Let $i, j$ be such that $s = \langle i, j\rangle$. Let $\tau = T_{5s+2}(\emptyset) = T_{5i}(\sigma)$. By conditions (iii) and (iv), $\sigma$ is of some even length $2n$. If $\left|W_{h_j(n)}\right| \leq p_i(n)$, find $t \leq p_i(n)$ such that $t \notin W_{h_j(n)}$ (given $h_j$ and $p_i$, this is computable in $\mathbf{0}'$), and let $T_{5s+3} = T_{5s+2}E_{r\frown 0}$ where $T_{5s+2}(r) \supseteq T_{5i}(\sigma \frown t)$ (such $r$ exists by condition (iv)). Then $T_{5s+3}$ satisfies conditions (i)-(iv) and fixes $f_i(n) = t$, thus forcing $R^2_s$.

If $\left|W_{h_j(n)}\right| > p_i(n)$, then $R^2_s$ is already satisfied. In this case, simply let $T_{5s+3} = T_{5s+2}$.

**Stage** $5s+3$:

(i) If there is some $x$ and some $\tau = T_{5s+3}(\sigma)$ which forces $\varphi_s^A(x)\uparrow$, let $x, \tau, \sigma$ be the first such with $|\sigma|$ even, and let $T_{5s+4} = T_{5s+3}E_\sigma$. This will force $\varphi_s^A$ to not be total.

(ii) Otherwise, if there is some $\tau = T_{5s+3}(\sigma)$ above which there are no $s$-splits, let $\tau, \sigma$ be the first such with $|\sigma|$ even, and let $T_{5s+4} = T_{5s+3}E_\sigma$. This will force $\varphi_s^A$ to be recursive by Lemma 6.1.5 (i).

(iii) Otherwise, if there is some $\tau = T_{5s+3}(\sigma)$ above which there are no $s$-splits which agree on the odds, let $\tau, \sigma$ be the first such with $|\sigma|$ even. The idea here is to construct a subtree of $T_{5s+3}E_\sigma$ with the branching at every even level being pairwise $s$-splitting.

We replace $T_{5s+3}, T_{5s+4}$ with $T, T'$, resp., for readability. The construction is by induction on the tree level, starting with $T'(\emptyset) = \tau$. Suppose we have defined level $n'$ of $T'$, consisting of strings from level $n$ of $T$ (where $n - n'$ is even). If $n'$ is odd, simply copy the successors from level $n+1$ of $T$, that is let

$$T'(\sigma \frown i) = T((T'/T)(\sigma) \frown i) \quad \text{for all } \sigma \in \mathrm{dom}(T') \text{ of length } n \text{ and } i = 0, 1.$$

Note that this is clearly in accordance with conditions (i)-(iv). If $n'$ is even, let $\tau_i = T'(\sigma_i)$ (for $1 \le i \le r$) be the strings on level $n'$ of $T'$, and let $q$ be the branching size at level $n$ of $T$. We go by induction on $i \le r$ defining a set of extensions $\rho_{i,j}$ for $j < q$. First let $\rho_{0,j} = T((T'/T)(\sigma_0) \frown j) - \tau_0$ (that is, the original $q$ extensions of $\tau_1$ on $T$). For the induction step, suppose we have defined $\rho_{i-1,j}$ for all $j < q$. We define strings $\rho_{i,j}^m$ by a subinduction on $m$ running over an ordered list of all pairs $\langle t, t' \rangle \in q \times q$ with $t < t'$. First let $\rho_{i,j}^{-1} = \rho_{i-1,j}$. Given $\rho_{i,j}^{m^-}$, with the next pair being $m = \langle t, t' \rangle$, let $(\chi_0, \chi_1)$ be the first $s$-split above $\tau_i \frown \rho_{i,t}^{m^-}$ on $T$, and let $x$ be the first such that $\varphi_s^{\chi_0}(x) \neq \varphi_s^{\chi_1}(x)$. Let $\chi_2$ be the first extension of $\tau_i \frown \rho_{i,t'}^{m^-}$ on $T$ such that $\varphi_s^{\chi_2}(x)\downarrow$. Let $\rho_{i,t'}^{m} = \chi_2 - \tau_i$, and let $\rho_{i,t}$ be $\chi_k - \tau_i$ where $k < 2$ is the first such that $(\tau_k, \tau_2)$ $s$-splits. Let $\rho_{i,j}^{m} = \rho_{i,j}^{m^-}$ for all other $j$. This completes the subinduction. Having defined $\rho_{i,j}^{m}$ for the last pair $m$, let $w$ be the least odd number not less than all levels of $T$ on which the strings $\tau_i \frown \rho_{i,j}^{m}$ lie. For all $j < q$, let $\rho_{i,j}$ be the least extension of $\rho_{i,j}^{m}$ such that $\tau_i \frown \rho_{i,j}$ lies on level $w$ of $T$. This completes the induction on $i$. Finally, let $T'(\sigma_i \frown j) = \tau_i \frown \rho_{r,j}$ for all $1 \leq i \leq r$ and $j < q$. This completes the construction in this case.

$T'$ satisfies conditions (i)-(iv). Furthermore, every branching in $T$ which disagrees on the odds $s$-splits, while no $s$-split agrees on the odds. Thus $T'$ is $s$-splitting on the odds, and consequently forces $\varphi_s^A \equiv \mathrm{Od}(A)$ by Lemma 6.1.5 (iii).

(iv) Otherwise, it is necessarily the case that there are $s$-splits agreeing on the odds above every node on $T_{5s+3}$. In this case we construct a fully $s$-splitting tree $T_{5s+4}$ with $T_{5s+4}(\emptyset) = T_{5s+3}(\emptyset)$. At even levels, we apply the same process of erecting splits which was applied in case (iii). At odd levels the process is similar, but with two differences. First, there is no subinduction on $m$, since there is only one pair ($q = 2$ here). Second, we need to define the extensions $\rho_{i,j}$ (for $j = 0, 1$) to agree on the odds. The latter slightly complicates the definition of $\rho_{i,j}$ given $\rho_{i-1,j}$: Let $(\chi_0, \chi_1)$ be the first $s$-split above $\tau_i \frown \rho_{i-1,0}$ on $T$, which agrees on the odds, and let $x$ be the first such that $\varphi_s^{\chi_0}(x) \neq \varphi_s^{\chi_1}(x)$. Let $\chi_2$ be the first extension of $\chi_0[\tau_i \frown \rho_{i-1,0} \to \tau_i \frown \rho_{i-1,1}]$ on $T$ such that $\varphi_s^{\chi_2}(x)\downarrow$. Let $k < 2$ be the least such that $(\chi_k, \chi_2)$ is an $s$-split (which agrees on the odds, note). Now let $\rho_{i,0}$ and $\rho_{i,1}$ be extensions of equal length of $\rho^*_{i,0} = \chi_k - \tau_i$ and $\rho^*_{i,1} = \chi_2 - \tau_i$ (resp.), which agree on the odds, by copying the extension from the longer to the shorter. For instance, if $\rho^*_{i,0}$ is the longer, let $\rho_{i,0} = \rho^*_{i,0}$ and let $\rho_{i,1} = \rho^*_{i,0}[(\rho^*_{i,0} \upharpoonright |\rho^*_{i,1}|) \to \rho^*_{i,1}]$.

The tree thus created forces $A \leq \varphi^A_s$, by Lemma 6.1.5 (ii).

In each of these cases we get a uniform subtree $T_{5s+4}$ satisfying conditions (i)-(iv), which forces $R^3_s$.

**Stage** $5s+4$**:** The goal here is to build a subtree of $T_{5s+4}$, fat enough so as to have $> p_{s+1}(m)$-branching at level $2m$ (for every $n$).

The tree $T' = T_{5s+5}$ is constructed by induction on levels. As before, $T_{5s+4}$ is denoted by $T$. First let $T'(\emptyset) = T(\emptyset)$. Suppose we have defined level $n'$ of $T'$, consisting of strings from level $n$ of $T$ (where $n - n'$ is even). If $n'$ is odd, copy the successors from level $n+1$ of $T$, as was done in case (iii). If $n' = 2m$ is even, let $\tau$ be a string on level $n$ of $T$, and let $w$ be the least odd number such that there are $q > p_{s+1}(m)$ successors of $\tau$ on level $w$ of $T$, name them $\tau \frown \rho_j$ for $j < q$. For each $\sigma$ of length $n'$ such that $T'(\sigma)\downarrow$, and for each $j < q$, let $T'(\sigma \frown j) = T'(\sigma) \frown \rho_j$. This ends the construction.

$T_{5s+5}$ satisfies conditions (i)-(iv), and has $> p_{s+1}(n)$-branching at level $2n$, as required by the construction.

$\square$

# Chapter 7

# A Non-Weakly Recursive Minimal Below $\mathbf{0}'$

We answer a question raised in [Ish99]. An outline is given in section 1, and the construction itself in section 2.

## 7.1 Introduction

We want to construct a minimal degree below $\mathbf{0}'$ which is not weakly recursive. In order to achieve that, we apply the diagonalization method presented in Chapter 6 to the classic construction of a minimal degree below $\mathbf{0}'$, using partial recursive trees and a finite injury argument [Ler83, IX.2.1].

Since $\mathbf{0}'$ is the oracle, we do not have the privilege of enumerating the recursive functions. However, if $p_i$ (or $h_i$) is not total that fact will be found out eventually, and we can go back and act accordingly, just as we do when we run out of splittings.

As any minimal degree is not a.n.r. (see Theorem 1.2.2(i)), this result tightens the bound on the scope of Ishmukhametov's characterization. The structure of $\Delta^0_2$ degrees (i.e. $\mathcal{D}(\leq \mathbf{0}'))$ is considered to be the "closest", in many aspects, to the structure of recursively enumerable Turing degrees. Here we see that, while every r.e. degree is either weakly recursive or array non-recursive, the same does not hold for the $\Delta^0_2$ degrees.

## 7.2 The Construction

**Theorem 7.2.1.** *There is a minimal degree* $\mathbf{a} < \mathbf{0}'$ *which is not weakly recursive.*

*Proof.* The requirements here are:

$R_k^0$ : $\quad A \neq \varphi_k$

$R_k^1$ : $\quad k = \langle i, j\rangle \wedge \varphi_i, \varphi_j$ are total $\rightarrow \exists n \ \left(\left|W_{\varphi_j(n)}\right| > \varphi_i(n) \ \vee \ f_i(n) \notin W_{\varphi_j(n)}\right)$

$R_k^2$ : $\quad \varphi_k^A = C$ is total $\rightarrow (C \leq_T \mathbf{0} \ \vee \ A \leq_T C)$

The construction is a finite injury one, with oracle $\mathbf{0}'$. At stage $s$ we construct partial recursive trees $T_i^s$ for $i \leq g(s)$, and a string $\alpha_s = T_{g(s)}^s(\emptyset)$. Following the notation of Chapter 6, we use $\tau \sqsubset_i^s \tau'$ to mean that $\tau'$ is an immediate successor of $\tau$ on $T_i^s$. Since a single degree is being constructed, no uniformity is needed here. We will satisfy the following conditions for all $s$ and $i \leq g(s)$:

(i) If $i = 4k$ and $\varphi_k(n)\downarrow$ then every non-terminal node on the $n$-th level of $T_i^s$ has $> \varphi_k(n)$ immediate successors.

(ii) If $i = 4k+1$ then $T_i^s$ forces $R_k^0$.

(iii) If $i = 4k+2$ then $T_i^s$ forces $R_k^1$ ($f_i$ are defined below).

(iv) If $i = 4k+3$ then $T_i^s$ is either a $k$-splitting subtree or a non-splitting extension subtree of $T_{i-1}^s$.

(v) If $\tau_1$ is a non-terminal node on $T_i^s$ and $\tau_1 \sqsubset_{i'}^s \tau_2$ for some $i' < i$, then there exists some $\tau_3 \supseteq \tau_2$ such that $\tau_1 \sqsubset_i^s \tau_3$.

For each $i$, the trees $T_i^s$ will approach a partial recursive limit tree $T_i$. The function $A = \cup\alpha_s$ is the unique path lying on all of the limit trees. The functions

$f_i$ are computed as follows:

$$f_i(n) = C(n) \quad \text{where } A = T_{4i}(C).$$

We start with $g(0) = 0$ and $T_0$ being the identity tree with domain $\{\sigma : \sigma(n) \leq \varphi_0(n)\}$ (we assume wlog that $\varphi_0$ is total).

At stage $s+1$, let $g(s+1) = g(s)+1$ (this might be changed if there is an injury). Let $T_i^{s+1} = T_i^s$ for all $i \leq g(s)$. Let $T = T_{g(s)}^{s+1}$. We define $T' = T_{g(s)+1}^{s+1}$ by cases as described below. Before carrying out the appropriate instructions, however, we check for convergence of all computations $\varphi_i(x)$ which are mentioned (only finitely many, so the $\mathbf{0}'$ oracle suffices). If $\varphi_i(x)$ is found to diverge, $\varphi_i$ is marked non-total.

(i) If $g(s+1) = 4k+1$, let $T' = T \cdot E_r$ where $r$ is the first such that $T(r)$ is incompatible with $\varphi_k$ (this is computable in $\mathbf{0}'$). This can fail if $T(\emptyset)$ is terminal on $T$, in which case an injury occurs (see below).

(ii) If $g(s+1) = 4k+2$, let $i, j$ be such that $k = \langle i, j \rangle$. If either of $\varphi_i, \varphi_j$ has been found to be non-total in a prior stage, there is nothing to be done, so let $T' = T$. Otherwise observe that $\alpha_s = T(\emptyset) = T_{4i}^{s+1}(\sigma)$ for some $\sigma$, and let $n = |\sigma|$. Look for the least $p \leq \varphi_i(n)$ such that $p \notin W_{\varphi_j(n)}$ (a $\mathbf{0}'$ computation). If $p$ is found, find the $q$ such that $T(q) \supseteq T_{4i}^{s+1}(\sigma \frown p)$ (exists by property (v) above), and let $T' = T \cdot E_q$. If no such $p$ is found then $|W_{\varphi_j(n)}| > \varphi_i(n)$, and the requirement is already fulfilled, so let $T' = T$. The same applies to the case where $\varphi_j(n)\uparrow$. If, however, $T(\emptyset)$ is terminal on $T$, an injury occurs (see below). Note that this includes the cases where $\varphi_i(n)\uparrow$ or $\alpha_s$ is terminal on $T_{4i}^{s+1}$.

(iii) If $g(s{+}1) = 4k{+}3$ let $T'(\emptyset) = T(\emptyset)$, and define $T'$ to be a $k$-splitting subtree of $T$ in a manner which preserves property (v): when we have defined $T'(\sigma') = T(\sigma) = \tau$ which has $m$ immediate successors on $T$, we define immediate successors $T'(\sigma' \frown p) \supseteq T(\sigma \frown p)$ for all $p < m$, which $k$-split pairwise. The process is much simpler than in Chapter 6, as we do not need to have uniformity: start with $\rho_p = T(\sigma \frown p)$ for $p < m$, then for each pair $u < v < m$ extend $\rho_u$ and $\rho_v$ to be $k$-splitting (and still on $T$). If this process cannot be completed, $\tau$ is terminal on $T'$.

(iv) If $g(s+1) = 4k+4$ let $T'(\emptyset) = T(\emptyset)$, and define $T'$ to satisfy condition (v). When we have defined $T'(\sigma') = T(\sigma) = \tau$ with $|\sigma'| = n$, let $\{\tau_p\}_{p<m}$ be the first set of pairwise incompatible successors of $\tau$ on $T$ such that $m > \varphi_{k+1}(n)$ and every immediate successor of $\tau$ on $T$ is contained in some $\tau_p$. Let $T'(\sigma' \frown p) = \tau_p$ for $p < m$. If no such set of successors is found, $\tau$ is terminal on $T'$.

In cases (i) or (ii) above, injury occurs if $\alpha_s = T(\emptyset)$ is terminal on $T$. In this case, let $j$ be the least such that $\alpha_s$ is terminal on $T_j^s$. Note that $j$ is either $4k+3$ or $4k+4$, since in the other two cases the tree is an extension subtree, and cannot be the first on which $\alpha_s$ is terminal. In either case, let $g(s+1) = 4k+3$, and as before let $T_i^{s+1} = T_i^s$ for all $i < g(s+1)$. If $j = 4k+3$ then some string $\tau \supseteq \alpha_s$ on $T_{4k+2}^s$ has no $k$-splits above it on that tree. On the other hand, if $j = 4k+4$, then there are no infinite paths extending $\alpha_s$ on $T_{4k+3}^s$ (the existence of such path would yield a set of successors as required in case (iv) above). This in turn means that some extension of $\alpha_s$ is terminal on $T_{4k+3}^s$, and again the existence of such a string $\tau$ is implied. Find the first such $\tau = T_{4k+2}^s(\sigma)$ using $\mathbf{0}'$, and set $T_{4k+3}^{s+1} = T_{4k+2}^{s+1} \cdot E_\sigma$, a non-splitting extension subtree. This ends the construction.

Now we turn to the proof. The trees reach a limit by the usual argument: once $T_j^s$ have stabilized for all $j < i$, $T_i^s$ can be injured at most once (if $i = 4k + 3$). The sequence $\alpha_s$ is increasing, and so $A = \cup\alpha_s$ is a function recursive in $\mathbf{0}'$, the oracle for the construction. The requirements $R_k^0, R_k^1, R_k^2$ are forced by the trees $T_{4k+1}, T_{4k+2}, T_{4k+3}$ (respectively), and so $\mathbf{a} = \deg(A)$ is minimal and not weakly recursive (refer to Lemma 6.1.5). $\square$

# Chapter 8

# Very Weakly Recursive Degrees

We define very weakly recursive degrees, and modify Ishmukhametov's original proof to show that this possibly larger class of degrees has the strong minimal cover property as well. At this point we have not been able to determine if there exists a very weakly recursive degree which is not weakly recursive.

**Definition 8.1.** A degree $\mathbf{a}$ is *very weakly recursive* if there is a function $p \leq \mathbf{a}$ such that for every function $f \leq \mathbf{a}$ there is a recursive function $h$ with $|W_{h(n)}| \leq p(n)$ and $f(n) \in W_{h(n)}$ for all $n \in \omega$.

Clearly every weakly recursive degree is very weakly recursive.

**Theorem 8.2.** *Every very weakly recursive degree* $\mathbf{a}$ *has a strong minimal cover.*

*Proof.* Let $A \in \mathbf{a}$, and let $p$ be the recursive function given by Definition 8.1.

We construct a set $M$ with $M(2n) = A(n)$, such that the following requirements are satisfied:

$$N_e : \quad \varphi_e^M \text{ is total } \rightarrow (\varphi_e^M \leq_T A \vee M \leq_T \varphi_e^M),$$

$$P_e : \quad M \neq \varphi_e^A.$$

In order to construct $M$, we will construct a sequence of trees $\{T_e\}_{e\in\omega}$ such that for all $e \in \omega$:

(i) $T_e \leq_T A$ is a full binary tree,

(ii) $T_{e+1}$ is a subtree of $T_e$, with $T_{e+1}(\emptyset) \supset T_e(\emptyset)$,

(iii) $T_{e+1}$ forces both $N_e$ and $P_e$.

We will then define $M$ as the unique path which lies on all of the trees $T_e$. $M$ will consequently satisfy all requirements $N_e$ and $P_e$, and will thus be a strong minimal cover for $A$.

We start by defining

$$T_0 = \{\sigma : \forall n\ (\sigma(2n)\downarrow \rightarrow \sigma(2n) = A(n))\}$$

(that is, the full tree of "$A$-true" strings). $T_0$ clearly satisfies condition (i).

At stage $e > 0$ the trees $T_k$ for $k \leq e$ have been already defined and satisfy conditions (i)-(iii). There are two cases.

*Case* 1. There are no $e$-splits on $T_e$ above some $\sigma = T_e(\tau)$.

In this case let $T_{e+1} = T_e \cdot E_{\tau^*}$, where $\tau^* \supset \tau$ is the first such that $T_e(\tau^*) \not\subseteq \varphi_e^A$. Conditions (i) and (ii) are clearly satisfied by $T_{e+1}$. For condition (iii), note that $N_e$ is forced since if $\varphi_e^M$ is total then it can be computed from the non-$e$-splitting $T_{e+1}$.

*Case* 2. $e$-splits are dense on $T_e$.

In other words, there is an $e$-split above every $\sigma \in \mathrm{rg}(T_e)$. Merely taking the splitting subtree as $T_{e+1}$ will not do here, as it will produce a minimal cover which is not necessarily strong.

Therefore, in addition to the tree $T_{e+1}$ we shall also construct a partial tree $T : \omega^{<\omega} \rightarrow 2^{<\omega}$ for which the following conditions hold:

(a) $T$ is partial recursive (i.e. has a r.e. graph),

(b) $T$ is $e$-splitting (i.e. every branching $e$-splits),

(c) $T_{e+1} \subseteq T$.

Given such $T$ and $T_{e+1}$, it is easy to show that $T_{e+1}$ forces the requirement $N_e$. Indeed, if $M$ is a path on $T_{e+1}$, and $\varphi_e^M = C$ is total, then we can compute $M$ from $C$ by enumerating $\mathrm{rg}(T)$ and throwing away all strings $\sigma$ for which $\varphi_e^\sigma(n)$ is incompatible with $C$. Condition (b) guarantees that the remaining strings are all compatible, while condition (c) guarantees that they will form the full set $M$, which has arbitrarily long initial segments in $\mathrm{rg}(T_{e+1})$.

For the purpose of constructing $T$, we will first introduce some terminology.

**Definition 8.3.** (i) An *$e$-splitting $k$-fan* is a set $F = \{\tau_0; \tau_1, \ldots, \tau_k\}$ of distinct strings, such that $\tau_i$ and $\tau_j$ form an $e$-split of $\tau_0$ for every $1 \leq i < j \leq k$. $\tau_0$ is the *root* of $F$ and $\tau_1, \ldots, \tau_k$ are the *blades* of $F$.

(ii) Two $e$-splitting fans $F_0, F_1$ *$e$-split a string* $\sigma$ if their roots $e$-split $\sigma$.

We now define a function $f$ recursively in $A$. Let $\sigma^* = T_e(\tau^*)$ be the first non-root string incompatible with $\varphi_e^A$ (this may not be computable in $A$, but it is only done once, and no uniformity is claimed). Find two $e$-splitting $p(1)$-fans $F_0^{\sigma^*}, F_1^{\sigma^*} \subseteq \mathrm{rg}(T_e)$ which split $\sigma^*$, and let $f(0) = [F_0^{\sigma^*} \cup F_1^{\sigma^*}]$, where $[\cdot]$ here denotes a recursive coding of finite sets of strings (not to be confused with the notation for the set of paths on a tree).

Inductively, suppose we have defined

$$f(n-1) = \left[ \bigcup_{i<t_n, j<2} F_j^{\sigma_i} \right]$$

where $F_0^{\sigma_i}, F_1^{\sigma_i} \subseteq \mathrm{rg}(T_e)$ are $e$-splitting $p(n)$-fans which $e$-split $\sigma_i \in \mathrm{rg}(T_e)$. Find for each blade $\tau$ of every fan $F_j^{\sigma_i}$, two $p(n+1)$-fans $F_0^\tau, F_1^\tau \subseteq \mathrm{rg}(T_e)$ which split $\tau$, and let

$$f(n) = \left[ \bigcup_{\tau,j} F_j^\tau \right]$$

This definition can be carried out with oracle $A$ since both $T_e$ and $p$ are computable in $A$, and splitting fans will always be found because of the density of $e$-splits on $T_e$.

By Definition 8.1, there is a recursive function $h$ such that

$$\forall n \in \omega \; (f(n) \in W_{h(n)} \;\wedge\; |W_{h(n)}| \leq p(n)).$$

Now we construct $T$. First throw into $T$ all strings in $\{\sigma^*\} \cup F_0^{\sigma^*} \cup F_1^{\sigma^*}$ (that is, let $\sigma^*$ be the root of $T$, the two fan roots its successors, and the all the blades as their successors on level 2 of $T$). Call $F_0^{\sigma^*}$ and $F_1^{\sigma^*}$ *accepted fans of level 0.* Now simultaneously enumerate all $W_{h(n)}$ for $n > 0$, and for each element $a^n = [U]$ enumerated in $W_{h(n)}$ do the following:

**First validation:** Check that $U$ is a union of $e$-splitting $k$-fans for some fixed $k$, and that all strings in $U$ agree on even elements. If not, drop $a^n$.

**Second validation:** Wait until some $a^{n-1}$ is enumerated in $W_{h(n-1)}$ and passes the first and second validation, such that for each blade of a fan given by $a^{n-1}$ there is a distinct pair of fans given by $a^n$ which split it. If this happens, then $a^n$ passes the second validation, and is said to be *extending* $a^{n-1}$.

Now, label the elements of $W_{h(n)}$ as $a_1^n, \ldots, a_{k_n}^n$ in order of enumeration, with $k_n \leq p(n)$. For each accepted fan $F$ of level $n-1$, given by some $a^{n-1}$, with blades $\tau_1, \ldots, \tau_r$, and for each $1 \leq i \leq r$, if $a_i^n$ exists and extends $a^{n-1}$, then enumerate in $T$ the two fans given by $a_i^n$ which split $\tau_i$ (setting their roots to be successors of $\tau_i$ in $T$), and call them *accepted fans of level $n$*. This ends the construction of $T$.

Properties (a),(b) clearly hold. Next we define $T_{e+1}$ by induction as a subtree of $T$, to ensure property (c). Note that for all $n > 0$ the set coded by $f(n)$ will pass both validations. Set $T_{e+1}(\emptyset) = \sigma^*$ and set $T_{e+1}(i)$ to be the root of $F_i^{\sigma^*}$ $(i = 0, 1)$. Suppose we have defined level $n > 0$ of $T_{e+1}$, which consists of roots of accepted fans of level $n-1$ given by $f(n-1)$. Given $\tau$ on level $n$, the root of the accepted fan $F$, there is a blade $\tau'$ of $F$ such that there are two accepted fans $F_0^{\tau'}$ and $F_1^{\tau'}$ given by $f(n)$, which $e$-split $\tau'$ (and hence $\tau$). This is because $F$ is a "true" fan, which has no more than $p(n)$ blades. Let the roots of the fans $F_i^{\tau'}$ be the successors of $\tau$ on level $n+1$.

Now, $T_{e+1}$ is a subtree of $T$ since all roots (as well as all blades) of accepted fans appear on $T$. It is also a subtree of $T_e$, since all roots (as well as all blades) of "true" fans appear on $T_e$. It was constructed from $T$ with knowledge of $f$, so it is computable in $A$. It is clearly $e$-splitting, hence forces $P_e$. Finally, it forces $N_e$ and satisfies condition (ii) by choice of $\sigma^*$. $\square$

# Chapter 9

# Constructing a Non-Join

We give a direct construction of a non-join, with the hope that it may be built upon to produce interesting results (e.g. concerning the class of cuppable degrees).

**Lemma 9.1.** *There are degrees* $0 < \mathbf{b} < \mathbf{a} \leq \mathbf{0}''$ *such that no degree* $\mathbf{c} < \mathbf{a}$ *satisfies* $\mathbf{b} \vee \mathbf{c} = \mathbf{a}$.

*Proof.* The requirements which we are trying to satisfy are

$$
\begin{aligned}
R^0_s: &\quad \mathrm{Od}(A) \neq \varphi_s \\
R^1_s: &\quad s = \langle e_1, e_2 \rangle \rightarrow \\
&\quad \left(\varphi^A_{e_1} \text{ is total } \wedge \varphi_{e_2}^{\varphi^A_{e_1} \oplus \mathrm{Od}(A)} \text{ is total and equal to } A \rightarrow \varphi^A_{e_1} \geq_T A\right)
\end{aligned}
$$

We construct a sequence of recursive uniform binary trees $T_i$ such that for all $i \in \omega$ and binary strings $\sigma$ we have $T_{i+1} \subseteq T_i$ and

$$|\sigma| \text{ is odd } \leftrightarrow \mathrm{Od}(T_i(\sigma \frown 0)) = \mathrm{Od}(T_i(\sigma \frown 1)).$$

That is, the levels of each tree alternate between agreeing and disagreeing on the odds.

To define $T_0$, let

$$T_0(d_0d_1d_2d_3\ldots) = (0d_0)\frown(d_10)\frown(0d_2)\frown(d_30)\frown\ldots$$

for all $d_i < 2$ (the parenthesis are for readability).

At stage $2s$, let $\sigma$ be the first of length 2 such that $\mathrm{Od}(T_{2s}(\sigma))$ is incompatible with $\varphi_s$. Let $T_{2s+1} = T_{2s}E_\sigma$. This tree is in the required form, and satisfies $R^0_s$.

At stage $2s+1$, let $T = T_{2s+1}$, and let $e_1, e_2$ be such that $s = \langle e_1, e_2\rangle$. We construct $T' = T_{2s+2}$ as follows.

(i) If there is some $\sigma$ and $x$ such that $T(\sigma)$ forces $\varphi^A_{e_1}(x)\uparrow$ on $T$, let $\sigma$ be the first such of even length, and let $T' = T\cdot E_\sigma$.

(ii) Otherwise, if there is some $\sigma$ and $x$ such that $T(\sigma)$ forces $\varphi_{e_2}^{\varphi^A_{e_1}\oplus\mathrm{Od}(A)}(x)\uparrow$ on $T$, let $\sigma$ be the first such of even length, and let $T' = T\cdot E_\sigma$.

(iii) Otherwise, if there is some $\sigma$ and $x$ such that

$$\varphi_{e_2}^{\varphi^{T(\sigma)}_{e_1}\oplus\mathrm{Od}(T(\sigma))}(x)\downarrow\neq T(\sigma)(x)$$

let $\sigma$ be the first such of even length, and let $T' = T\cdot E_\sigma$.

(iv) Otherwise, we first prove

**Proposition 9.2.** *Every $\sigma$ has extensions $\sigma_0, \sigma_1$ such that $T(\sigma_0), T(\sigma_1)$ agree on the odds and $e_1$-split.*

*Proof.* Assume wlog that $|\sigma|$ is odd, and let $x$ be the first on which $T(\sigma\frown 0)$ and $T(\sigma\frown 1)$ disagree (note that $x$ is even, since these strings must agree on the odds). Since cases (i),(ii) do not hold, there is some $\sigma_0 \supseteq \sigma\frown 0$ such that

$\varphi_{e_2}^{\varphi_{e_1}^{T(\sigma_0)} \oplus \mathrm{Od}(T(\sigma_0))}(x)\downarrow$. Let $\sigma_1^* = \sigma_0[\sigma \frown 0 \to \sigma \frown 1]$. Similarly, there is some $\sigma_1 \supseteq \sigma_1^*$ such that $\varphi_{e_2}^{\varphi_{e_1}^{T(\sigma_1)} \oplus \mathrm{Od}(T(\sigma_1))}(x)\downarrow$.

Now, $T(\sigma_0)$ and $T(\sigma_1)$ agree on the odds by uniformity of $T$ and because $\mathrm{Od}(T(\sigma \frown 0)) = \mathrm{Od}(T(\sigma \frown 1))$. On the other hand, we must have

$$\varphi_{e_2}^{\varphi_{e_1}^{T(\sigma_i)} \oplus \mathrm{Od}(T(\sigma_i))}(x)\downarrow = T(\sigma_i)(x)$$

for both $i = 0, 1$, since case (iii) fails. But $T(\sigma_0)(x) \neq T(\sigma_1)(x)$ by choice of $x$, and so the oracles must disagree. Since $\mathrm{Od}(T(\sigma_0)) = \mathrm{Od}(T(\sigma_0))$, it must be that $\varphi_{e_1}^{T(\sigma_0)}$ is incompatible with $\varphi_{e_1}^{T(\sigma_1)}$, hence $T(\sigma_0)$ and $T(\sigma_1)$ form an $e_1$-split. $\square$

The proposition implies that we can construct an $e_1$-splitting subtree of $T$ in the required form. The construction is by induction on levels, and is similar in nature to the one carried out in case (iv) of the proof of Theorem 6.2.1. Start with $T'(\emptyset) = T(\emptyset)$. Suppose we have defined level $n$ of $T'$, and let $\tau_i = T'(\sigma_i)$ be the strings on that level (for $1 \leq i \leq r$). We define by induction on $i \leq r$ strings $\rho_{0,i}, \rho_{1,i}$, starting with $\rho_{j,0} = \emptyset$ for $j = 0, 1$. Having defined $\rho_{j,i-1}$, find a $e_1$-split $\chi_0, \chi_1 \supseteq \tau_i \frown \rho_{0,i-1}$ on $T$, such that $\mathrm{Od}(\chi_0) = \mathrm{Od}(\chi_1)$ (the existence of such is guaranteed by the proposition). Let $x$ be the first such that $\varphi_{e_1}^{\chi_0}(x)\downarrow \neq \varphi_{e_1}^{\chi_1}(x)\downarrow$, and let $\chi_2 \supseteq \chi_0[\tau_i \frown \rho_{0,i-1} \to \tau_i \frown \rho_{1,i-1}]$ on $T$ such that $\varphi_{e_1}^{\chi_2}(x)\downarrow$ (guaranteed to exist since case (i) fails). Let $k < 2$ be the first such that $\varphi_{e_1}^{\chi_k}(x) \neq \varphi_{e_1}^{\chi_2}(x)$. Now let $\rho_{i,0}$ and $\rho_{i,1}$ be extensions of equal length of $\rho_{i,0}^* = \chi_k - \tau_i$ and $\rho_{i,1}^* = \chi_2 - \tau_i$ (resp.), which agree on the odds, by copying the extension from the longer to the shorter. For instance, if $\rho_{i,0}^*$ is the longer, let $\rho_{i,0} = \rho_{i,0}^*$ and let $\rho_{i,1} = \rho_{i,0}^*[(\rho_{i,0}^* \upharpoonright |\rho_{i,1}^*|) \to \rho_{i,1}^*]$. Note that $\mathrm{Od}(\rho_{i,0}) = \mathrm{Od}(\rho_{i,1})$. This completes the induction on $i$. Finally, if $n$

is odd, simply let $T'(\sigma_i \frown k) = \tau_i \frown \rho_{r,k}$ for all $1 \leq i \leq r$ and $k = 0, 1$. This is a uniform $e_1$-splitting branching which agrees on the odds. If $n$ is even we need to make the branching disagree on the odds, so let $\rho_0, \rho_1$ be the first extensions of equal length of $\rho_{r,0}, \rho_{r,1}$ (resp.), which disagree on the odds, such that $\tau_1 \frown \rho_k \in \mathrm{rg}(T)$ for $k < 2$ (such exist since the branching on every even level of $T$ disagrees on the odds). Let $T'(\sigma_i \frown k) = \tau_i \frown \rho_k$ for all $1 \leq i \leq r$ and $k < 2$. Thus level $n+1$ of $T'$ is constructed, and the induction on $n$ is complete.

Note that $T_{2s+2} = T'$ is an $e_1$-splitting tree, and therefore it forces requirement $R^1_s$ by Lemma 6.1.5.

In each of the cases, $T_{2s+2}$ is a tree of the required form which forces $R^1_s$. Now let $A$ be the unique path lying on all $T_i$, and let $B = \mathrm{Od}(A)$. Since the requirements $R^0_s$ are satisfied, $B$ is not recursive. Since the requirements $R^1_s$ are satisfied, if $X \leq_T A$ and $X \oplus B \geq_T A$ then $X \geq_T A$. It follows that $\mathbf{a} = deg(A)$ and $\mathbf{b} = \deg(B)$ are as stated in the lemma. $\square$

# Bibliography


[Coo73] S. B. Cooper. Minimal degrees and the jump operator. *J. Symbolic Logic*, 38:249–271, 1973.

[DJS96a] Rod Downey, Carl G. Jockusch, and Michael Stob. Array nonrecursive degrees and genericity. In *Computability, enumerability, unsolvability*, volume 224 of *London Math. Soc. Lecture Note Ser.*, pages 93–104. Cambridge Univ. Press, Cambridge, 1996.

[DJS96b] Rod Downey, Carl G. Jockusch, and Michael Stob. Array nonrecursive degrees and genericity. In *Computability, enumerability, unsolvability*, volume 224 of *London Math. Soc. Lecture Note Ser.*, pages 93–104. Cambridge Univ. Press, Cambridge, 1996.

[DLS96] Rodney G. Downey, Steffen Lempp, and Richard A. Shore. Jumps of minimal degrees below $\mathbf{0}'$. *J. London Math. Soc. (2)*, 54(3):417–439, 1996.

[Eps79] Richard L. Epstein. *Degrees of unsolvability: structure and theory*, volume 759 of *Lecture Notes in Mathematics*. Springer, Berlin, 1979.

[Fri57] Richard Friedberg. A criterion for completeness of degrees of unsolvability. *J. Symb. Logic*, 22:159–160, 1957.

[Ish99] Shamil Ishmukhametov. Weak recursive degrees and a problem of Spector. In *Recursion theory and complexity (Kazan, 1997)*, volume 2 of *de Gruyter Ser. Log. Appl.*, pages 81–87. de Gruyter, Berlin, 1999.

[KHNS] Bjørn Kjos-Hanssen, André Nies, and Frank Stephan. On a question of Ambos-Spies and Kučera. In preparation.

[Lac68] A. H. Lachlan. Distributive initial segments of the degrees of unsolvability. *Z. Math. Logik Grundlagen Math.*, 14:457–472, 1968.

[Ler83] Manuel Lerman. *Degrees of unsolvability.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1983. Local and global theory.

[Sac63] Gerald E. Sacks. Recursive enumerability and the jump operator. *Trans. Amer. Math. Soc.*, 108:223–239, 1963.

[Sho78] Richard A. Shore. On the $\forall\exists$-sentences of $\alpha$-recursion theory. In *Generalized recursion theory, II (Proc. Second Sympos., Univ. Oslo, Oslo, 1977)*, volume 94 of *Stud. Logic Foundations Math.*, pages 331–353. North-Holland, Amsterdam, 1978.

[Sho88] Richard A. Shore. A noninversion theorem for the jump operator. *Ann. Pure Appl. Logic*, 40(3):277–303, 1988.

[Sim77] Stephen G. Simpson. First-order theory of the degrees of recursive unsolvability. *Ann. of Math. (2)*, 105(1):121–139, 1977.

[Spe56] Clifford Spector. On degrees of recursive unsolvability. *Ann. of Math. (2)*, 64:581–592, 1956.

[SS99] Richard A. Shore and Theodore A. Slaman. Defining the Turing jump. *Math. Res. Lett.*, 6(5-6):711–722, 1999.