The Turing Degrees: Global and Local Structure [1]

Richard A. Shore ©

May 20, 2015

[1] THIS IS A ROUGH DRAFT. CORRECTIONS, COMMENTS AND SUGGGESTIONS ARE WELCOME. PLEASE SEND TO SHORE@MATH.CORNELL.EDU. PLEASE INCLUDE THE DATE OF THE DRAFT. THANKS.

Contents

preface

Preface

....Thank previous students and especially Mia Minnes who took notes in TeX for the entire course in 2007 along with the other students that term who all took turns preparing lecture notes.....

intro

Introduction

The introduction

Reference Style to Chapters Chapter n Sections §n.m??

notation

0.1 Notation

$\mathbb{N}$ $a, b, c, d, e, i, j, k, l, m, n, r, s, t, u, v, w, x, y, z$

$\mathbb{N} \to \mathbb{N}$ f, g, h

$\mathbb{N} \to 2$ sets $A, B, C, U, V, W, X, Y, Z$

f ambiguous

partial functions ϕ, φ, ψ...

Functionals Φ, Ψ, ... (continuous)

0.2 Pairing and ordered sequences

Choice. Uniformity.

specific pairing polynomial $\frac{1}{2}(x^2 + 2xy + y^2 + 3x + y)$ then n-tuples by recursion. then picture for listing the elements of a countable family of countable sets.

Pairing functions: desiderata for $\langle x, y \rangle$.

$2^x 3^y$; $\frac{1}{2}(x^2 + 2xy + y^2 + 3x + y)$

$\langle x, y, z \rangle = \langle x, \langle y, z \rangle \rangle$ etc.

$\langle x_1, \ldots x_n \rangle = \langle n, \langle x_1, \langle x_2 \ldots \rangle \rangle \rangle$

Uniformity over length n.

$\prod p_i^{x_i+1} - 1$ for $\langle x_1, \ldots x_n \rangle$

pairing for functions $f \oplus g$, $\oplus_i f_i$

$\oplus A_i$ $\oplus f_i$ $\oplus \{A_i | ...i...\}$

strings $\alpha, \beta, \gamma, \delta, \rho, \sigma, \tau$

String notation functional form if $\sigma = \langle x_1, \ldots x_n \rangle$ then $\sigma(i) = x_{i+1}$ (perhaps prefer $\langle x_0, \ldots x_{n-1} \rangle$); $\mathrm{dom}(\sigma) = n = |\sigma|$ length of σ; order by initial segments $\sigma \subseteq \tau$; restriction for $m \leq |\sigma|$, $\sigma \restriction m \subseteq \sigma$ and $|\sigma \restriction m| = m$. Apply to functions on all of $\mathbb{N}$ as well: $\sigma \subseteq f$; $f \restriction m$. so strings as finite sequences

Notation: set of all finite sequences of elements of S denoted by $S^{<\omega}$ set of sequences of length n by S^n

binary strings $\{0,1\}^n$ $\{0,1\}^{<\omega}$

Identify $\{0,1\}$ with 2 and more generally $\{0,1,\ldots,n-1\}$ with n and so write 2^n, $2^{<\omega}$ $2^{\mathbb{N}}$ ...

Also pairing for strings...

Chapter 1

overview

An Overview

hist

1.1 History, intuition, undecidability, formalization, ...

formal

1.2 Formal definitions, models of computation

Turing machines (multitape with input, output and others)

n-ary functions formally view as given by input an n-tuple $x_1, \ldots, x_n$ coded on same input tape e.g. as $B1^{x_1}B1^{x_2} \ldots B1^{x_n}B \ldots$

??or in some nice way e.g. machine $\langle e, n \rangle$ ... or just realize can mimic such functions ...and use notation of n-ary inputs as shorthand.?

Also allow multiple oracles $f_1, \ldots f_n$ ...

Other notions: prim recursive + μ (search); register machines; equation calculus?..

relcomp

1.3 Relative computability: Turing machines with oracles

define Turing machines, input (note how to deal with n-tuples as input coded on same on tape) above, output, computation also for oracles (functions) , Turing reducibility $\leq_T$

define formally

do for explicitly for unary functions

can do n-ary in same way or by coding n-tuples as numbers

also oracles - could have tuple of oracles or view as code by function pairing

ref specific coding procedures in .

Church-Turing Thesis

equality for partial functions $\Phi = \Psi$ for if either are defined at x then both are defined and their values are equal.

use here example

Proposition 1.3.1 *Turing reducibility is reflexive and transitive.*

Recursive. (continuous) Functionals; oracles as inputs

degrees

1.4 Degrees, types of questions

We have seen that Turing reducibility, $\leq_T$, is a reflexive and transitive relation and so we can consider the equivalence classes for this relation, i.e. the classes of the form $\{g|g \leq_T f \,\&\, f \leq_T g\}$ for any function f. These classes are called the Turing degrees, or simply the degrees. As, by Exercise 1.4.1, every Turing degree contains a set (i.e. a characteristic function) and so we may, without any loss consider just the degrees of sets or the classes $\{B|B \leq_T A \,\&\, A \leq_T B\}$. We denote the collection of all degrees by $\mathcal{D}$. We typically denote the degree containing f or A by $\mathbf{f}$ or $\mathbf{a}$. The class $\mathcal{D}$ is a partial order under the induced relation $\leq$ defined by $\mathbf{f} \leq \mathbf{g} \Leftrightarrow f \leq_T g$. (See Exercise 1.4.2.)

.

degset **Exercise 1.4.1** *For every function $f \in \mathbb{N}^{\mathbb{N}}$ there is a set $A \in 2^{\mathbb{N}}$ of the same Turing degree, i.e. $f \leq_T A$ and $A \leq_T f$.*

welldef **Exercise 1.4.2** *The relation $\leq$ defined above is well defined, reflexive and transitive on the degrees and so makes them into a partial order.*

Notation 1.4.3 *For functions $f, g \in \mathbb{N}^{\mathbb{N}}$, we write $f|_T g$ to denote that f and g are Turing incomparable, i.e. $f \not\leq_T g$ and $g \not\leq_T f$. Similarly, we write $\mathbf{f}|\mathbf{g}$ when $\mathbf{f} \not\leq \mathbf{g}$ and $\mathbf{g} \not\leq \mathbf{f}$.*

outline book
algebraic, local...
second order, global: definability, automorphisms, theory
mention appendices

Chapter 2

basics

The Basics

codeTM

2.1 Coding Turing machines with oracles

To aid in our analysis of Turing machines with oracles and relative computability as described in refcomp 1.3, we introduce a master (universal) recursive function in two forms. First we have

$$\varphi(f,e,x,s)=y.$$

Here the variables are f a function, e a number (index), x a number (input), s a number (steps) and the expression means that the Turing machine with index e and oracle j given input x and run for s many steps converges and outputs y. Secondly we have an approximation version

$$\varphi(\sigma,e,x,s)=y$$

where the variables are σ a string (so an initial segment of a function), e a number (index), x a number (input), s a number (steps, use); and the expression means that the Turing machine with index e and oracle restricted to σ given input x and run for s many steps converges and outputs y. We say that $\varphi(f,e,x,s)$ or $\varphi(\sigma,e,x,s)$ converges, if there is such a y. The standard notation for this is $\varphi(f,e,x,s)\downarrow$ or $\varphi(\sigma,e,x,s)\downarrow$.

??(In later chapters it will at times be convenient to allow σ to be more generally a finite partial function from $\mathbb{N}$ to $\mathbb{N}$ rather than restricting its domain to be an initial segment of $\mathbb{N}$. This makes essentially no difference in our discussions here.)??

The specific formal definition of φ is not important. We now make explicit some properties that we want it to have. The properties should be obvious natural ones based on our descriptions of the action of Turing machines with oracles.

??redo following as definitions etc.??

Properties of $\varphi(\sigma,e,x,s)$:

(i) **Use:** If $\sigma\subseteq\tau$ and $\varphi(\sigma,e,x,s)\downarrow=y$ then $\varphi(\tau,e,x,s)=y$

(ii) **Permanence:** If $s<t$ and $\varphi(\sigma,e,x,s)\downarrow=y$ then $\varphi(\sigma,e,x,t)\downarrow=y$

(iv) **Computable Domain:** The domain of φ is computable, in other words there is a procedure to decide whether φ converges on any given tuple (σ, e, x, s). This procedure simply runs the machine with index e on input x and oracle σ. If the machine arrives at an output by step s, then answer yes (and otherwise, answer no).

We next adopt some conventions.

Conventions: We convert these partial functions into total ones by adopting the convention that, if, during the run of the machine with index e and oracle f or σ on input x, the computation does not halt in s steps we output $*$ (some designated reserved symbol). In the case of a finite oracle σ, we also output $*$ if we ask a question of the oracle for a number outside the domain of σ. For later convenience we also adopt the conventions that the output is $*$ if the input x or the expected output y is greater than s.

These conventions can be thought of as saying that, in s many steps, we cannot compute anything about any $x > s$ (so roughly we have to read the input first) and that we cannot ask any oracle questions about $z > s$ either (so roughly new must be able to write the number z down before asking about the value of $f(z)$) and cannot compute an output $y > s$ (so roughly we have write the output on the tape).

Finally,

We next note that φ is universal in the sense that it allows us to uniformly capture all the partial Turing functions with oracles.

Proposition 2.1.1 *For any function f and natural number e, $\Phi_e^f(x) = y$ if and only if $\exists\sigma \subseteq f \exists s\big[\varphi(\sigma, e, x, s)\downarrow= y\big]$. Thus $f \leq_T g$ if and only if $\exists e(\Phi_e^g = f)$.*

Definition 2.1.2 *We define the* use of the computation *of $\Phi_e^f(x) = y$ as the least n such that $\varphi(f \restriction n+1, e, x, s) = y$. We also say that $\sigma = f \restriction n$ is the* axiom *(about the oracle f) that gives this computation. Note that if f is changed at or below the use then this axiom no longer applies to the changed function and while there may be a convergent computation from the new function, it is not the same computation giving the output y. ??have to define* the computation *...??*

Definition 2.1.3 *$\Phi_{e,s}^f(x) = y$ means that if we run Φ_e^X on x for s steps we get y as our result in the sense that x and $\varphi(f \restriction s, e, x, s) = y$. Similarly, we use $\Phi_e^\sigma(x) = y$ to mean that $x < |\sigma|$ and $\varphi(\sigma, e, x, |\sigma|) = y$. So that the additional **convention** here is that when the oracle is a finite string σ we run the machine for $|\sigma|$ many steps. Thus we use $\Phi_e^\sigma(x)$ to denote what we might also write as $\Phi_{e,|\sigma|}^\sigma(x)$. With this convention in mind we write $\varphi(\sigma, e, x)$ for $\varphi(\sigma, e, x, |\sigma|)$.*

We collect our various notations in the following remark.

Remark 2.1.4 $\Phi_e^f(x) = y \Leftrightarrow \exists s \varphi(f,e,x,s) = y \Leftrightarrow (\exists \sigma \subset f)(\exists s)(\Phi_{e,s}^\sigma(x) = y) \Leftrightarrow$
$(\exists \sigma \subset f)(\varphi(\sigma,e,x,s) = y) \Leftrightarrow (\exists \sigma \subset f)(\Phi_e^\sigma(x) = y) \Leftrightarrow$
$(\exists \sigma \subset f)(\varphi(\sigma,e,x) = y)$.
$\Phi_e^f(x) \downarrow \Leftrightarrow \exists s \varphi(f,e,x,s) \downarrow \Leftrightarrow (\exists \sigma \subset f)(\exists s)(\Phi_{e,s}^\sigma(x) \downarrow) \Leftrightarrow$
$(\exists \sigma \subset f)(\varphi(\sigma,e,x,s) \downarrow) \Leftrightarrow (\exists \sigma \subset f)(\Phi_e^\sigma(x) \downarrow) \Leftrightarrow$
$(\exists \sigma \subset f)(\varphi(\sigma,e,x) \downarrow)$
$f \leq_T g \Leftrightarrow \exists e(\Phi_e^g = f)$.

Definition 2.1.5 *We use Φ_e to denote $\Phi_e^\emptyset$ the eth Turning machine with oracle the empty set (constant function 0). This is equivalent (explain) to the list of Turing machines without oracles and we often simply identify these two versions.*

Remark 2.1.6 (Multivariable Functions) *Just as we extended unary functions given by basic Turing machines (with or without oracles) to multivariable functions in* mvtm **??**, *we can extend the universal partial recursive functions and predicates to their analogs for multiple inputs. Thus we have $\varphi_n(f,e,\bar{x},s) = y$, $\varphi_n(\sigma,e,\bar{x},s) = y$, $\varphi_n(f,e,\bar{x},s) \downarrow$ and $\varphi_n(\sigma,e,\bar{x},s) \downarrow$ which are as above except that the input is now an n-tuple $\bar{x}$ in place of a unary x.*

Remark 2.1.7 (Relativization) *As usual we can add extra function parameters $\bar{h}$ by relativization and so define $\varphi_n(f,\bar{h},e,\bar{x},s) = y$, $\varphi_n(\sigma,\bar{h},e,\bar{x},s) = y$, $\varphi_n(f,\bar{h},e,\bar{x},s) \downarrow$ and $\varphi_n(\sigma,\bar{h},e,\bar{x},s)\downarrow$ which are as above except that the functions and predicates are recursive in $f \oplus \bar{h}$, $\bar{h}$, $f \oplus \bar{h}$ and $\bar{h}$, respectively.*

transtred **Theorem 2.1.8** *Turing reducibility is transitive, i.e. if $f \leq_T g$ and $g \leq_T h$ then $f \leq_T h$.*

Proof. ?? ■

We now state some classical theorems about basic manipulations of Turing machines and their indices. They were originally presented with formal proofs based on various formal definitions of computable functions. Now we tend to view them as "obvious" based on our approach that assumes the Church-Turing thesis.

smn **Theorem 2.1.9 (s-m-n Theorem)** *For each $m,n \in \mathbb{N}$, there is a one-one recursive function s_n^m of such that*

$$\forall f \forall \bar{y} \big[\Phi_e^f(x_1,\ldots,x_m,y_1,\ldots,y_n) = \Phi_{s_n^m(e,\bar{x})}^f(\bar{y})\big].$$

In fact, can view m and n as variables as well and so have a single recursive function s such that .??Decide on approach to n-ary functions in previous Chapter??

Informal Proof. ■

Notion of uniformity here or before
typically formally given by s-m-n theorem

ujoin **Example 2.1.10 (Uniformity of Join)** *There is a recursive function p such that,for every f, e and i, $\Phi_e^f \oplus \Phi_i^f = \Phi_{p(e,i)}^f$.*

utrans **Example 2.1.11 (Uniformity of Transitivity)** *There is a recursive function t such that,for every f, e and i, $\Phi_e^{\Phi_i^f} = \Phi_{t(e,i)}^f$.*

padding **Theorem 2.1.12** *Padding Lemma: $\forall e \exists^\infty i \forall f (\Phi_e^f = \Phi_i^f)$.*

Proof. Exercise: Informal argument and formal one using s-m-n Theorem. ■

Idea that s-m-n gives more: uniformity.

enumthm **Theorem 2.1.13** *Enumeration Theorem: List of partial recursive (in f) functions: Φ_e^f.*

recthm **Theorem 2.1.14 (Recursion Theorem aka Fixed Point Theorem)** *If f is a recursive function then there is an e such that for all g, $\Phi_e^g = \Phi_{f(e)}^g$.*

Proof. ■

This is Kleene's essentially one-line seemingly magical proof of the Recursion Theorem. A different perhaps more reasonably seen as discoverable is given below as an application of Theorem dhrnotrec 3.0.19.

Intuitively the recursion theorem implies that we can call a function h within the definition of h itself. This may seem counterintuitive or simply false. But think of the procedure that we envision defining h except that it has calls to h. Replace the calls to f by calls to Φ_e. this gives us a computation procedure whose index (as a Turing machine) is clearly recursive in e. Let f be the function computing the index of this procedure. By the recursion theorem f has a fixed point e. Now argue that Φ_e is a function (at least a partial function) as desired for h.

The Recursion Theorem is also used when we talk about approximation procedures.

Generalizations, relativizations?

Relativization in general and specifically here e.g. for universal φ and conventions (need for forcing language)

Issue of multiple inputs and oracles. formally via pairing informally write out as sequence?

Chapter 3

TD

The Turing Degrees

We have defined the basic notion of relative complexity of computation on functions, $f \leq_T g$ in ??. We saw that it is a transitive relation and so we defined the equivalence classes as the Turing degrees, $\mathbf{f} = \{g | f \leq_T g \ \& \ g \leq_T f\}$. We then have the partial order $\leq$ on $\mathcal{D}$, the set of Turing degrees, induced by $\leq_T$. We now want to present some simple but important facts about the structure of $\mathcal{D}$ that can be deduced from what we know already.

facts

Facts about the Turing Degrees:

1. $\mathcal{D}$ is a partial order under $\leq_T$.

2. $\mathcal{D}$ has a least element, $\mathbf{0}$, which is the degree containing all computable sets.

3. $\mathcal{D}$ is an uppersemilattice, i.e. there is a join operation $\vee$ on degrees such that, for every pair of degrees $\mathbf{f}, \mathbf{g}$, $\mathbf{f} \vee \mathbf{g}$ is their least upper bound. It is the degree of $f \oplus g$:

4. There are elements of $\mathcal{D}$ other than $\mathbf{0}$. There are two types of proofs of this fact. One is a counting argument: There are $2^{\aleph_0}$ sets (subsets of $\mathbb{N}$). By Cantor's theorem $2^{\aleph_0} > \aleph_0$ i.e. there are uncountably many sets. Moreover, since there are only countably many Turing machines, each degree is at most countable as a set. (In fact, as $f \equiv_T f + c$ any constant function c, each degree contains) and has at most countably many predecessors. So, not only are there degrees other than $\mathbf{0}$, there are $2^{\aleph_0}$ many degrees. Other specific ones given later (e.g. DNR functions and the Halting Problem to start refs??.)

- There is no largest degree because for each degree $\mathbf{x}$, can find a DNR relative to $\mathbf{x}$ and it is not below it. Relativization

- $\mathcal{D}$ is an upper semi-lattice.
  We can define a join operator $\vee$ on $\mathcal{D}$: On sets/functions it is defined by

$$f \oplus g(2n) = f(n) \qquad\qquad f \oplus g(2n+1) = g(n);$$

or $(f \oplus g)(\langle n, 0\rangle) = ...$

this is inherited by the degrees $\mathbf{f} \vee \mathbf{g} =$ degree of$(f \oplus g)$. Note that we can use the join operator to produce a degree strictly above each degree because can take join of a member of the degree with some DNR relative to it. Also by counting: take any g not recursive in f (only countably many) and consider $f \oplus g$.

Note that we denote degrees in boldface, **a** or **f** and sets or functions in lightface, A or f.

We summarize these facts as follows.

Theorem 3.0.15 *$\mathcal{D}$ is an uppersemilattice with 0 of size $2^{\aleph_0}$ with the countable predecessor property.*

In (??) we see that every countable partial order and even uppersemilattice can be embedded in $\mathcal{D}$. This also holds for ones of size $\aleph_1$ (??). Indeed each is isomorphic to an initial segment (downward closed subset) of $\mathcal{D}$. For these results $\aleph_1$ is as far as we can go. There are models of ZFC in which $2^{\aleph_0} > \aleph_1$ with uppersemilattices (partial orders) of size $\aleph_2$ that cannot be embedded in (as initial segments of) $\mathcal{D}$. ??

Exercise 3.0.16 *There is a cofinal sequence of degrees if and only if CH (continuum hypothesis) holds in which case the sequence can be chosen to have order type $\aleph_1$.*

graph **Exercise 3.0.17** *Every degree contains a set (i.e. characteristic function). (Graph(f))*

So can use sets or functions indiscriminately as oracles when defining degrees...sometimes technical advantages to working with one or the other...

??Note that we identify sets with their characteristic functions so when we talk about functions f, g ... we include the possibility that they are sets. ??

Some more questions about $\mathcal{D}$: how tall is it? how wide is it? is it a lattice? ... Answers coming up.....

How do we "build" a nonrecursive function. We can "implement" the idea of the proof of Cantor's theorem that there are more functions on $\mathbb{N}$ than elements of $\mathbb{N}$, i.e. $2^{\aleph_0} > \aleph_0$. This idea is a really a procedure called a diagonal argument.

We extract the crucial property in our setting in the following definition.

Definition 3.0.18 (DNR) *A function h is DNR (diagonally non-recursive) if $\forall n(h(n) \neq \Phi_n(n))$.*

dnrnotrec **Proposition 3.0.19** *If h is DNR then h is not recursive.*

Proof. By the diagonal argument... ■

Relativize definition and proposition

We can now prove the recursion theorem.

Proof of Recursion theorem. Suppose not, i.e. $\forall e(\Phi_e \neq \Phi_{f(e)})$. [Such an f is called fix point free (FPF).] We try to build a recursive DNR h for the desired contradiction.

Since $\Phi_e(e) \neq \Phi_{f(e)}$ for every e, we only need to make $\Phi_{h(e)} = \Phi_{f(\Phi_e(e))}$ to get $h(e) \neq \Phi_e(e)$. "Obviously" (by the $s{-}m{-}n$ theorem), there is such a recursive h: given e find the index of the machine which first computes $\Phi_e(e)$ and if it converges then computes f of the value and begins mimicking the machine with that index. This gives the description a machine that computes $\Phi_{f(\Phi_e(e))}$ and so an index, $h(e)$ for it. Going from e to $h(e)$ is an intuitively computable procure. Formally, the s-m-n theorem shows that it is a recursive function. On the other hand, our assumption (that f is FPF) implies (as above) that h is DNR for the desired contradiction. ■

Now to recover the standard constructive version of the theorem that actually computes the fixed point (with the usual uniformity), note that the index k for h can be found recursively in that of f (again by the $s-m-n$ theorem). Now $\Phi_{h(e)} = \Phi_{\Phi_k(e)} = \Phi_{f(\Phi_e(e))}$ and so if we let $e = k$ then $h(k) = \Phi_k(k)$ is the desired fixed point: $\Phi_{h(k)} = \Phi_{\Phi_k(k)} = \Phi_{f(\Phi_k(k))}$.

Chapter 4

re

R.E. Sets and the Turing Jump

4.1 The Jump Operator

defjump **Definition 4.1.1** *On functions, define the jump as $f' = \{e : \Phi_e^f(e)\downarrow\}$.*

Proposition 4.1.2 *$f \leq_T g$ implies that $f' \leq_T g'$ and so the* jump operator *is well-defined on the degrees.*

Proof. Since $f \leq_T g$, there is i such that $f = \Phi_i^g$. So

$$e \in f' \Leftrightarrow \Phi_e^f(e)\downarrow \Leftrightarrow \Phi_e^{\Phi_i^g}(e)\downarrow$$

The s-m-n theorem gives a recursive one-one function k such that $\Phi_{k(e,i)}^g = \Phi_e^{\Phi_i^g}$. In particular,

$$e \in f' \Leftrightarrow \Phi_{k(e,i)}^g \downarrow \Leftrightarrow k(e,i) \in g'$$

and so $f' \leq_T g'$. Thus if $f \equiv_T g$ then $f' \equiv_T g'$ and the jump operator is well defined on $\mathcal{D}$. ■

Proposition 4.1.3 *$f \leq_T f'$*

Proof. We need to compute $f(n)$ using f'?

If f is the characteristic function of a set A then we can decide whether $n \in A$ using A' by recursively finding an e such that $\Phi_e^A(e)\downarrow \Leftrightarrow n \in A$. Formally, we can appeal to the s-m-n theorem to get a recursive one-one function k such that for each n, $\Phi_{k(n)}^A(k(n))\downarrow \Leftrightarrow n \in A$. This gives the desired reduction.

We can now appeal to Exercise graph 3.0.17 for the theorem for all functions f. Or we can prove it directly. ■

Exercise 4.1.4 *Give a direct proof that $f \leq_T f'$ for all functions. Solution: Instead of finding index of machine which asks whether $n \in A$, find a machine such that*

$$\Phi_{k(n,m)}^f(z)\downarrow \Leftrightarrow f(n) = m.$$

Then successively ask $k(n, 0) \in f'$?, $k(n, 1) \in f'$?, etc. until find $k(n, m) \in f'$ in which case output $f(n) = m$. This procedure halts because of the assumption that f is a function, hence total.

Proposition 4.1.5 $f <_T f'$

Proof. It clearly suffices to find an $h \leq_T f'$ which is DNR^f, i.e.

$$\forall n\big(h(n) \neq \Phi_n^f(n)\big)$$

We compute h from f' as follows: Given n, ask if $\Phi_n^f(n)\downarrow$ (in other words, if $n \in f'$). If so, let $y = \Phi_n^f(n)$ and put $h(n) = y + 1$. If not, set $h(n) = 0$. ■

Conclusion: The jump is a strictly increasing, order preserving operator on the degrees.

The jump of the empty set is, of course, $\emptyset'$. By our identification of the Turing machines without oracles with those with oracle $\emptyset$, it is identified with the usual halting problem $K = \{e | \Phi_e(e)\downarrow\}$, the set of indices e of Turing machines which halt on input e. One often wants to consider alternate versions such as $K_0 = \{\langle x, y\rangle\, | \Phi_x(y)\downarrow\}$. We can consider this as an alternative version of K or of the jump in general because the produce sets of the same degree.

Exercise 4.1.6 *For every f, $f' \equiv_T \{\langle x, y\rangle\, | \Phi_x^f(y)\downarrow\}$.*

In fact, more is true as we shall see in ?? (1-1 equivalence) and ?? (recursive isomorphism).

4.2 Trees and König's Lemma

So we have two ways of "getting" nonrecursive sets - diagonalization and the halting problem. Have seen that the second computes an example of the first. What about the other way? Does every DNR function compute K? If not, what can we say about the needed complexity (if there is any)? We take a side trip to an example of reverse mathematics and a comparison of the "strength" of versions of a well known combinatorial principle: König's Lemma.

While there are many mathematical definitions of a tree (and others are used later), for now we take a simple representation. Remembering that we are in the world of the natural numbers, it makes sense to use (for now at least) the following definitions.

tree **Definition 4.2.1** *A* tree *T is a subset of $\mathbb{N}^{<\mathbb{N}}$, the set of finite strings of natural numbers, that is closed downward under the natural partial order $\sigma \subseteq \tau$: σ is an initial segment of τ. (Or in the functional notation $\sigma(n) = \tau(n)$ for every $n < |\sigma|$. (We use $|\sigma|$ to denote the length of the sequence σ or in the functional notation its domain with the ordering on $\mathbb{N}$ given by $\in$ on the usual set theoretic representations of the natural numbers.) The root* *of any of our trees is then the empty string (or set) $\emptyset$. A* binary tree *is a tree of binary sequences, i.e. a downward closed subset of* $2^{<\mathbb{N}}$. *A* finitely branching tree *is a tree T such that for every $\sigma \in T$ there are only finitely many $\tau \in T$ with $\tau \supset \sigma$ and $|\tau| = |\sigma| + 1$. If T is a tree we say that a subset P of T is a* path on T *if P is infinite, linearly ordered and downward closed (with respect to $\subseteq$). The set of paths on T is denoted by $[T]$. The elements of a tree are often called* nodes *and ones with no successors in the tree,* leaves

Exercise 4.2.2 *If you know some general abstract definition of a (binary, finitely branching) tree, do all of ours satisfy the definition you know?*

Exercise 4.2.3 (Thought Problem) *Think about what a "converse" might mean. We are restricted to countable sets (trees) but can we think of any countable tree as ("isomorphic to") one of ours? In general, what does it mean to code mathematical structures in $\mathbb{N}$?*

KL **Lemma 4.2.4 (König's Lemma)** *If T is an infinite, finitely branching tree then T has an infinite path.*

Proof. We "construct" a path P in T by recursion. At each step t we have a node σ_t in T of length t with infinitely many successors on T. We begin, of course, with the root $\emptyset = \sigma_0$ relying on the fact that T is infinite to satisfy our condition. If we have σ_t we consider its immediate successors in T. By assumption there are only finitely many and so one of them say $\sigma_t\hat{\ }x$ has itself infinitely many successors on T. We let z be the least such x and let $\sigma_{t+1} = \sigma_t\hat{\ }z$. It is clear that $P = \{\sigma_t | t \in \mathbb{N}\}$ is a path in T. ■

WKL **Lemma 4.2.5 (Weak König's Lemma)** *If T is an infinite binary tree then T has an infinite path.*

Is this proof (of König's Lemma) constructive or effective? If not could there be one that is? Is it "easier" to prove Weak König's Lemma than the full one? Is it easier to construct a path in an infinite, binary tree than an arbitrary finitely branching one? What might these questions mean? Not every infinite tree has a path at all but what about arbitrary trees with paths? How hard is to construct one?

We begin with the first question. One way of making the question precise is to ask if every infinite finitely branching (or binary) recursive tree T has an infinite recursive path. Or more generally if every infinite finitely branching (or binary tree) T has an infinite path recursive in T. If so, we might also want there to be a uniformly effective procedure that produces such a path, i.e. an e such that $\Phi_e(T)$ is an infinite path in T for every finitely branching or perhaps every binary tree. The answer is no for all the versions and the proof is intimately connected to the notion of DNR functions. On the other hand, we claim that König's Lemma is more complicated than the weak version, i.e. it really is weaker. The analysis here is intimately connected to the jump operator.

Theorem 4.2.6 *There is an infinite recursive binary tree with no infinite recursive path.*

Proof. We want an infinite binary tree T such that for every $f \in [T]$, $f \in DNR$. If we did not have to make T recursive, we could simply take all the binary strings σ that satisfy the definition of a DNR function on their domains: $\{\sigma \in 2^{<\mathbb{N}} | (\forall n < |\sigma|)(\sigma(n) \neq \Phi_n(n))\}$. However, this set is not recursive (Exercise norectree ??). We can however, eventually recognize when a binary string σ fails to be in this set by seeing at some stage s that $\Phi_{n,s}(n) \downarrow = \sigma(n)$ for some $n < |\sigma|$. The picture for building the desired (or any) recursive tree is that we are effectively going along deciding which strings are in T. Say at stage s of our recursive construction we must decide for every binary string of length s if it is in T or not. (This makes T recursive.) We eliminate unwanted paths when we recognize that some σ has failed our test for being DNR. More precisely if at stage s we see that $\Phi_{n,s}(n) \downarrow = \sigma(n)$ for some $n < |\sigma|$ then no strings $\tau \supseteq \sigma$ are ever put into T at any stage $t \geq s$. Formally, $T = \{\sigma \in 2^{<\mathbb{N}} | (\forall n < |\sigma|)[\neg(\Phi_{n,|\sigma|}(n) \downarrow = \sigma(n)]\}$. By our basic facts about our master function φ, T is clearly recursive. Consider now any $f \in [T]$. If $f \notin DNR$ then there is some n and s such that $\Phi_{n,s}(n) \downarrow = f(n)$. By definition no $\sigma \subseteq f$ with $|\sigma| \geq n, s$ can be on T and so,of course, $f \notin [T]$ for the desired contradiction. ■

nonrectree **Exercise 4.2.7** *The tree* $S = \{\sigma \in 2^{<\mathbb{N}} | (\forall n < |\sigma|)(\sigma(n) \neq \Phi_n(n))\}$ *is not recursive.*

Theorem 4.2.8 *There is an infinite recursive finitely branching tree T such that every path in T computes $0'$.*

Proof. We want to code $0'$ into every infinite path f on a recursive tree T. Now T is a subset of $\mathbb{N}^{<\mathbb{N}}$ the set of all finite strings. In analogy with the previous construction, we might think of ourselves as beginning with the nonrecursive tree consisting of the single path f such that $f(n) = 0$ if $\Phi_n(n) \uparrow$ and $f(n) = s$ if s is the first stage t such that $\Phi_{n,t}(n) \downarrow$. We now want to turn this into a recursive, finitely branching tree T such that f is its only path. We follow the plan of keeping "bad" strings from extending to paths of the last construction and set $T = \{\sigma \in \mathbb{N}^{<\mathbb{N}} | (\forall n < |\sigma|)(\Phi_{n,|\sigma|}(n) \uparrow \;\Rightarrow\; \sigma(n) = 0 \;\&\; \Phi_{n,|\sigma|}(n) \downarrow \;\Rightarrow\; \sigma(n) = s$ where $\Phi_{n,s}(n) \downarrow$ but $\Phi_{n,s-1}(n) \uparrow)\}$. Now T is easily seen to be recursive from our basic facts about φ. Moreover by definition for each n there are at most two numbers r such that $\sigma(n) = r$ for any $\sigma \in T$ (0 and the first stage t such that $\Phi_{n,t}(n) \downarrow$). Thus T is finitely branching.

T is also infinite as, by induction, for every $\sigma \in T$ either $\sigma\hat{\ }0 \in T$ or $\sigma\hat{\ }s \in T$ for s the first stage t such that $\Phi_{n,t}(n) \downarrow$ (and perhaps both). We now claim that the f defined above is the only path on T. Suppose $g \in [T]$ and consider $g(n)$ for any n. If $\Phi_n(n) \uparrow$ then for every $\tau \in T$ with $|\tau| > n$, we must have $\tau(n) = 0$ by the definition of T. On the other hand, if $\Phi_n(n) \downarrow$ then let s be the first stage t such that $\Phi_{n,t}(n) \downarrow$. Again by the definition of T, if $\tau \in T$ and $|\tau| > n, s$ then $\tau(n) = s$. As $g \upharpoonright n + s \in T$, we must have $g(n) = s$ as required. ■

So solving the problem of finding a path in any infinite recursive finitely branching tree provides a calculation of $0'$. Note that one might say that T is 2-branching but it is not a binary tree under our current definitions. This is perhaps somewhat mysterious but an important distinction as well shall see. There are at most two immediate successors of each σ but we cannot recursively bound what they might be.

By relativization if we can find a path in every finitely branching tree, we can compute the jump operator. What about binary trees? It is by no means obvious, and indeed requires several ideas, to provide a proof but this is not the case for finding paths in infinite binary trees. How can we make this precise. We can capture the idea that it is "possible" to always be able to solve one problem (such as finding paths in infinite binary trees) without being able to solve another (finding paths in infinite finitely branching trees) by using the notion of a model. We understand 'being able" to include the idea that if we have some f then we have any $g \leq_T f$ and similarly if we have both f and g then we have $f \oplus g$. We make this precise by saying that there is a class $\mathcal{C}$ of functions closed under $\leq_T$ (and $\oplus$) such that such that for every $T \in \mathcal{C}$ that is (the characteristic function of) an infinite binary tree then there is an $h \in \mathcal{C}$ which is a path in T. So in $\mathcal{C}$ we can solve the first problem. On the other hand, there is an infinite finitely branching tree $T \in \mathcal{C}$ for which there is no path in $\mathcal{C}$. Thus we have a "model" in which every infinite binary tree has a path but not every infinite finitely branching tree has one. The proof of these assertions are in ??.

In the other direction, as every binary tree is finitely branching, it is immediate that if every infinite finitely branching tree in $\mathcal{C}$ has a path then so does every infinite binary tree. Thus we can conclude that solving the problem of finding paths for infinite finitely branching trees is strictly harder than the analogous problem for binary trees. This result is intimately related to a similar claim about how hard it is to prove Lemmas KL 4.2.4 and WKL 4.2.5 in the sense of what axioms are needed for the proof. This is the subject of reverse mathematics. We will return to such issues at a few points in this book. Survey or introductory articles include...????. The basic text is Simpson ??

Relations with finding a DNR function: DNR$_2 \equiv$ FPF, DNR$_k$ but DNR weaker? An example of reverse mathematics. Arbitrary trees much harder.

some exercises

Finding solutions for König's Lemma, even for recursive trees, requires more than $0'$. This is an example where closure under solving two problems is equivalent but one can not get by with a reduction that (effectively) transforms a problem of one type into one of the so that any solution of the second computes one of the first.

Medvedev and Muchnik degrees. ...For later after do INF$\equiv 0''$"

Exercise 4.2.9 *Show that every infinite, finitely branching tree T has a path recursive in T''. Build a recursive tree such that any path computes $0''$.*

Exercise 4.2.10 *Show that not every infinite tree has a path.*

Exercise 4.2.11 *Relation to compactness, topological and logical.*

4.3 Recursively Enumerable Sets

We began with the notion of what it means for a set or function to be computable (recursive). We now want to consider a weaker notion. The idea is that, for a set A, while we might not be able to decide if $n \in A$, we might nonetheless be able to list its elements. That is we might have a recursive function whose values are the elements of A (assuming $A \neq \emptyset$). For such sets we have a recursive way of enumerating its elements: $f(0), f(1), \ldots, f(n), \ldots$. So if $x \in A$ we eventually find out by enumerating that fact when we get to $f(n) = x$ for some n. If $x \notin A$ we may never discover that fact. (If we could, A would be recursive by ??

Some discussion that recursive enumerability is really about sets and not arbitrary functions e.g. if try to enumerate a function f by enumerating its graph then actually f is recursive not so for sets... So typically talk about sets A, B... when discussing notions of recursive enumerability.

Definition 4.3.1 *The following equivalent conditions define the statement "$A \subseteq N$ is recursively enumerable (r.e.) in B":*

- A is the domain of a partial recursive in B function. Notation: $W_e^B = \operatorname{dom} \Phi_e^B$ and approximations $W_{e,s}^B$??
- A is the range of a partial recursive in B function.
- A is either the range of a total recursive in B function or is empty.
- A is either the range of a 1-1 recursive in B function or is finite.

Proof. Argue that all equivalent. explain dovetailing. ■

Theorem 4.3.2 *A is recursive in B if and only if both A and $\bar{A} = \mathbb{N} - \mathbb{A}$ are r.e. in B.*

Theorem 4.3.3 *If A is r.e. in B and $B \leq_T C$ then A is r.e. in C.*

Recall that $A' = \{e : \Phi_e^A(e) \downarrow\}$. So, A' is r.e. in A because it is the domain of the function that on input e runs the eth machine with oracle A with input e. We want to show that A' is the most complicated set r.e. in A in various precise ways.

We say that a set A is reducible to one B if there is some procedure that allows us to decide membership in A using membership in B. We have already met the most important and fundamental such reducibility that of Turing: $A \leq_T B$. We can compute the membership of A by asking questions about the membership of elements in B during the computation. It may adaptively determine which questions it asks based upon answers to previous questions. We now define some other notions of reduction which are stronger than that of Turing in the sense that they imply but are not, in general, implied by Turing reducibility. These reductions are also primarily intended to apply to sets.
Definition.

1. ***1-1 reducibility***($\leq_1$): $A \leq_1 B$ if there exists a one-one recursive function f such that $\forall x\ x \in A$ if and only if $f(x) \in B$.

2. ***m or many-one reducibility*** ($\leq_m$): Same as one-one reducibility except f is an arbitrary (so possibly man-one) recursive function.

3. ***truth-table reducibility*** ($\leq_{tt}$): $A \leq_{tt} B$ if there exists a recursive function f such that $f(x)$ is a propositional formula σ in variables $p_1, ..., p_k$ such that for all $x\ x \in A$ if and only if B satisfies σ. EXPLAIN $B \models \sigma \Leftrightarrow T_B \models \sigma$ where T_B the assignment of truth values to propositional variables determined by setting assignment to v_n to be $B(n)$.

4. ***weak truth-table reducibility*** ($\leq_{wtt}$): $A \leq_{wtt} B$ if there exists a recursive function f and a Turing machine Φ_e such that $\Phi_e^B = A$ and the use of computation in $\Phi_e^B(x)$ is at most $f(x)$ for all x. This is sometimes called ***bounded Turing reducibility*** ($\leq_{bT}$). EXPLAIN

5. bounded number of queries (so also for functions)

Note that we have the following:

$$A \leq_1 B \to A \leq_m B \to A \leq_{TT} B \to A \leq_{wtt} B \to A \leq_{tt} B \to A \leq_T B.$$

Intuitively, we can think of the truth table reduction as giving a Boolean function which when given the answer to the oracle queries, produce the final answer of the reduction. Note that all time bounded complexity classes are tt reductions

The first three reducibilities are total procedures in the sense that applied to any set they always produce a set as output. The final one is not. It is like a tt reduction but may be partial on some sets. In fact tt reducibility is characterized by its being total on all set inputs.

Theorem 4.3.4 (Nerode's Theorem) *$A \leq_{tt} B$ if and only if there is e such that $A = \Phi_e^B$ and Φ_e^X is a total (characteristic) function for every X.*

Proof. Since tt is total by definition, one direction is immediate. For the other direction, say Φ_e^X is total for all X and $\Phi_e^B = A$. What happens when we run $\Phi_e^X(n)$ for some unknown X? We can build a computation tree (explain ??) which branches (in two) whenever the program asks a question $m \in X$ with the branches corresponding to the possible answers 0 or 1 to this question. We terminate the tree when the Turing machine halts (when it gets the answers supplied along the route followed so far). Since the computation halts for every oracle X, all possible paths are are terminated so (using even Weak König's Lemma) the tree is finite. We can build a truth table that corresponds to this reduction (propositional variables encode branch points and return outputs at end of every path). This is effective and gives a truth table reduction from A to B. ■

Diagram

This theorem depends essentially on the fact that we restricted our attention to sets rather than all functions. One way of looking at this is that $2^{\mathbb{N}}$ is a compact space (Cantor space) but $\mathbb{N}^{\mathbb{N}}$ (Baire space) is not. (The paths through a binary tree form a closed (and so compact) set in Cantor space. Each node at which we terminate the tree determine an open set (all paths extending it). If they cover the space (no paths in the tree) then by compactness some finite subset of these open sets cover the space and so the tree is finite (all nodes are initial segments of one of the finitely many nodes determining the open sets that form the cover of the whole space. Another (equivalent) one related to our discussion in the last ?? section is that if we are dealing with binary trees (we branched to 0 or 1 depending on whether some number is in our set) then if every path terminates, the whole tree is finite. (The compactness of $2^{\mathbb{N}}$ is equivalent to WKL. (EXPLAIN). The theorem is not true?? for arbitrary functions in the oracle. They would allow for infinite branching in our computation tree and König's Lemma fails for arbitrary trees ($\mathbb{N}^{\mathbb{N}}$ is not compact).

We now want to show that $\emptyset'$ is the most complicated r.e. set. We could show that $A \leq_T \emptyset'$ for every r.e. set A but in view of these new reducibilities we have just defined we can hope for more.

Definition 4.3.5 *A set A is called an r-*complete set *for class C if A is in $\mathcal{C}$ and for every $B \in \mathcal{C}$, $B \leq_r A$.*

Proposition 4.3.6 *A' is 1-complete for the class of sets r.e. in A.*

Proof. We already know that A' is R.E. in A. So we only need to prove for all e, $W_e^A \leq_1 A'$. By definition, $x \in W_e^A$ iff $\Phi_e^A(x)\downarrow$. So, the s-m-n theorem gives a recursive one-one k such that $\Phi_e^A(x) = \Phi_{k(e,x)}^A(k(e,x))$. Hence, we have $x \in W_e^A$ iff $k(e,x) \in A'$. ■

Proposition 4.3.7 *If $B \leq_m A'$ then B is r.e. in A so for all A, B, $B \leq_{m(1)} A'$ if and only if B is r.e. in A.*

Proof. By definition of $\leq_m$, there is recursive f such that $x \in B$ implies $f(x) \in A'$ implies $\Phi_{f(x)}^A(f(x))\downarrow$. So, we can use the s-m-n theorem to get that $x \in B$ iff $\Phi_i^A(x)\downarrow$ for some i, hence $B = W_i^A$. The rest of the assertion then follows from the previous Proposition. ■

Myhill isomorphism theorem and why this strongest equivalence from viewpoint of recursion theory.

We have seen that $A \leq_T B$ implies $A' \leq_T B'$. Now we present a similar result that links Turing-reduction with m(1)-reduction.

Proposition 4.3.8 *$A \leq_T B \Leftrightarrow A' \leq_m B'$ and $A \leq_T B \Leftrightarrow A' \leq_1 B'$.*

Proof. We first prove that $A \leq_T B \Rightarrow A' \leq_1 B'$. So, we want to determine whether $\Phi_x^A(x)\downarrow$ by asking a membership question in B'. We claim that $\Phi_x^A(x)\downarrow$ iff $\Phi_{f(x)}^B(f(x))\downarrow$ for some recursive 1-1 function f. Why? because for each x, we can produce a machine with oracle B which ignores its input and computes $\Phi_x^A(x)$ by simulating the machine Φ_x^A and whenever it asks a question about A, compute A from B as given by assumption. This gives a recursive method for producing index $f(x)$, which can be made 1-1 by the Padding Lemma. (or use s-m-n)

Now we prove $A' \leq_m B' \Rightarrow A \leq_T B$. In contrast, it is *not* the case that $A' \leq_T B' \Rightarrow A \leq_T B$. (see ??) mention and reference future results from Chapter 5??

Recall that $A \leq_1 A'$ (hence $A \leq_m A'$) because A is r.e. in A (it is the domain of procedure with oracle A which returns yes if $x \in A$ and loops forever otherwise). Likewise, $\bar{A} \leq_1 A'$, hence $\bar{A} \leq_m A'$, because $\bar{A}$ is r.e. in A.

By earlier converse, $A \leq_m A' \leq_m B'$ implies A is r.e. in B and $\bar{A} \leq_m A' \leq_m B'$ implies $\bar{A}$ is r.e. in B. Since A is recursive in B iff $A, \bar{A}$ are both r.e. in B (Theorem ??), A is recursive in B. ■

limitlemma **Theorem 4.3.9 (Shoenfield Limit Lemma)** *$A \leq_T B' \Leftrightarrow \exists f \leq_T B$ such that $\forall x\big(A(x) = \lim_{s\to\infty} f(x,s)\big)$. Note that asserting that $\lim_{s\to\infty} f(x,s)$ exists means that $f(x,s)$ is eventually constant for fixed x.*

??Slogan: Effective in the jump just in case have eventually correct recursive approximation.

Proof. Say $A \leq_T B'$, in other words $A = \Phi_e^{B'}$ equating the set with the function means that the characteristic function of A is $\Phi_e^{B'}$. We want $f \leq_T B$ such that $\lim_{s\to\infty} f(x,s) = A(x)$. Certainly, $A(x) = \lim_{s\to\infty} \Phi_{e,s}^{B'}(x)$. This is recursive in B', but not in B. In order to make it recursive in B, we want to approximate the oracle B' recursively in B. Since $B' = \{e : \Phi_e^B(e)\downarrow\}$, $B'_s = \{e : \Phi_{e,s}^B(e)\downarrow\}$ is an approximation for B' recursive in B. In fact, $B' = \lim_{s\to\infty} B'_s$ because approximation changes at most once for each e.

We can approximate any W_e^B similarly by $W_{e,s}^B = \{n : \Phi_{e,s}^B(n)\downarrow\}$ and $\lim_{s\to\infty} W_{e,s}^B = W_e^B$. ??Extract notation??

So, define f by $A_s(x) = \Phi_{e,s}^{B'_s}(x) = f(x,s)$ (with the convention that if Φ has not answered by time s, return " no"). Then $f \leq_T B$. It remains to verify that $A(x) = \lim_{s\to\infty} f(x,s)$. Since $A = \Phi_e^{B'}(x)$, there is s such that $A(x) = \Phi_{e,s}^{B'}(x) = \Phi_{e,t}^{B'}(x)$ for all $t \geq s$. The computation of A only uses finite information about B, say $\sigma \subseteq B$. Moreover there is s_1 such that $B'_t(n) = B'(n)$ for all $n < |\sigma|$ (aka $B'_t \restriction |\sigma| = B' \restriction |\sigma|$) for all $t \geq s_1$, because of permanence and the properties of limits.

Conversely, suppose there is $f \leq_T B$ and $A = \lim_{s\to\infty} f(x,s)$. We want to show that $A \leq_T B'$. To find $A(x)$, we could start computing $f(x,0), f(x,1), f(x,2)\ldots$ and we know that eventually we get the right answer. But how do we know when to stop? By definition

$$\exists s \forall t > s\big(f(x,t) = f(x,s)\big)$$

and for this s, $A(x) = f(x,s)$. Define the following program recursive in B: $\Phi_{k(s)}^B(s)\downarrow$ iff $(\exists t > s)\big(f(x,s) \neq f(x,t)\big)$. Note that $\{s : \Phi_{k(s)}^B(s)\downarrow\} \leq_T B'$. We can apply the program iteratively: does f change after stage 0? If so, can find s_0 where it changes. Does it change after s_0? etc. This procedure halts because f is eventually constant (since it is a limit). ∎

In applications of the Limit Lemma, without loss of generality we adopt the convention that we consider only functions f for which $\forall x(f(x,0)=0)$.

Theorem 4.3.10 *A is r.e. in B iff there is $f \leq_T B$ such that for all x, $A(x) = \lim_{s\to\infty} f(x,s)$ and $f(x,s)$ changes at most once ($|\{s : f(x,s) \neq f(x,s+1)\}| \leq 1$).?? Standardize by starting with $f(x,0)=0$??*

Proof. If there is such an f, let $\Phi_e^B(x)$ be the program which searches for an s such that $f(x,s)=1$, and halts if it finds one. Then $A = \operatorname{dom}\Phi_e^B$ so A is r.e. in B. Conversely, if A is r.e. in B, then $A = \operatorname{dom}\Phi_e^B$ for some e. Let f be the function

$$f(x,s) = \begin{cases} 1 \text{ if } \Phi_{e,s}^B(x)\downarrow \\ 0 \text{ otherwise.} \end{cases}$$

Then $f \leq_T B$, $\lim_{s\to\infty} f(x,s) = A(x)$ and $|\{s : f(x,s) \neq f(x,s+1)\}| \leq 1$. ∎

A is difference of sets r.e. in B if $\exists f \forall x f(x,s)$ changes at most twice. Then $A = C_0 - C_1$ both r.e. in B.

Continuing in this fashion, get the difference hierarchy (Putnam-Gold hierarchy).

Definition 4.3.11 *A is an n-r.e. set if there is a recursive function f such that for all x, $A(x) = \lim_{s\to\infty} f(x,s) = A(x)$ and $|\{s : f(x,s) \neq f(x,s+1)\}| \leq n$. ?? Standardize by starting with $f(x,0)=0$??*

We can connect this definition with difference of r.e. sets: A is n-r.e. iff

$$A = \begin{cases} (((W_{e_1} - W_{e_2}) \cup W_{e_3}) \cdots) - W_{e_n} \text{ if } n \text{ is even} \\ (((W_{e_1} - W_{e_2}) \cup W_{e_3}) \cdots) \cup W_{e_n} \text{ if } n \text{ is odd,} \end{cases}$$

where $W_{e_1}, \ldots, W_{e_n}$ are r.e. sets.

omegare **Definition 4.3.12** *A is ω-r.e. if there are recursive functions f, g such that $A(x) = \lim_{s\to\infty} f(x,s)$ and $f(x,s)$ changes at most $g(x)$ many times.*

idre **Exercise 4.3.13** *Show that if A is ω-r.e. then there is a $B \equiv_T A$ which is ω-r.e. with at most x many changes at x for each x.*

Exercise 4.3.14 *Show that, for each $\alpha \leq \omega$, there are α-r.e. sets which are not β-r.e. for any $\beta < \alpha$. Hint: list all n-r.e.(for fixed n or the for all n uniformly) sets and diagonalize making only $n+1$ (finitely) many changes .*

wtt0' **Exercise 4.3.15** *Show X is ω-r.e. iff $X \leq_{tt} 0'$ iff $X \leq_{wtt} 0'$.*

Note that in general, *tt* reducibility does not coincide with *wtt* reducibility. What do we know so far about the reducibilities?

Proposition 4.3.16 $\leq_T \neq \leq_m$, $\leq_T \neq \leq_1$

Proof. If $X \leq_m A$ and $X \leq_m \bar{A}$ and A is r.e. then X is recursive. Why? $X \leq_m A$ and A r.e. implies that X is re; $X \leq_m \bar{A} \Leftrightarrow \bar{X} \leq_m A$ so $\bar{X}$ r.e. as well. However, $0', \overline{0'} \leq_T 0'$, and $0'$ r.e. but not recursive. So $\leq_T \neq \leq_m$ and $\leq_T \neq \leq_1$. ■

Exercise 4.3.17 *Show that* $1-1, m, tt, wtt, T$ *are all distinct reducibilities. Hint: for wtt and T make list of the reductions (applied to some finite oracle). How hard is it to do this? Try for something recursive in* $0'$ *and then diagonalize. wtt but not tt is too hard. again list total tt-functions but now build both A and B in stages. In A put in only* 0 *except when might diagonalize. In B put in sequence of* 1*'s of length the next e to diagonalize ending at a place where we diagonalize in A and then at least one* 0*. (Fill in A with* 0*'s until this point.) Then fill in B with* 0*'s until force convergence so decide what to put into A. So for x to be in A must have* $x \in B$ *and* $x+1 \notin B$ *then check* $B \restriction x$ *to see how many* 1*'s in the list ending at x, say it is e, then compute how many* 0*'s need to put into B to make* $\Phi_e(x)\downarrow$ *and find answer.* $A(x)$ *is the opposite.??ref or move to or repeat after finite extension method.*

The Ershov hierarchy extends the difference hierarchy into the transfinite. If we exhaust the recursive ordinals produce precisely all the sets recursive in $0'$.

Recursively inseparable sets. Gödel's incompleteness theorem.

One-one equivalence same as recursive isomorphism. Explain, prove.

Index sets. Rice's Theorem.

Closure operations for r.e. sets. Reduction and Separation on complementary class. Uniformity issues. Point out easy ones. Argue somewhere for some not possible. Recursion theorem applications. ref completeness results later.

Define r.e. degrees, REA use later

arithh

4.4 Arithmetic Hierarchy

Notion of language for first order arithmetic. Then for arithmetic. Tension between expressiveness and simplicity. For our purposes want language to be recursive (and so all typical syntactic properties are recursive) and each function and relation to be (uniformly) recursive (and so all quantifier free relations are recursive). On the other hand want to as much as possible to be expressible as "simply" as possible.

What at a minimum. Want say 0 then perhaps successor $s(x)$ and/or addition $x+y$. In what sense is addition definable from successor (by recursion; implicitly; second order)? We want to restrict definability to first order formulas. Note that multiplication, $x \cdot y$, is not definable from addition.

Presburger addition is decidable.

Peano arithmetic or even Robinson arithmetic is not. Gödel's incompleteness theorem. (forward reference to proof). Idea of representation of recursive functions so decidability would solve the Halting problem. So we need at least multiplication. Typically put in $<$ and 1 as well although they are definable from addition.

Exercise 4.4.1 *Define $<$ and 1 from $+, 0$ in arithmetic.*

May want to put in more to make all recursive functions easily definable. Want quantifier free formulas to be uniformly recursive, however:

Exercise 4.4.2 *With a recursive language (and interpretation as uniformly recursive functions and predicates) it is not possible to define all recursive functions by quantifier free formulas.*

So we need to go to formulas with at least one quantifier. We can make life simple by adding in one master recursive predicate for $\varphi(\sigma, e, x, s) = y$ (so capturing the partial function). It is then immediate that every recursive predicate and function is definable by an existential formula, i.e. one of the form $\exists x_1 \exists x_2 \ldots \exists x_n \theta$ where θ is quantifier free. Or we can cite the theorem of Matijasevich (Davis, Putnam and Robinson) solving Hilbert's 10th problem negatively by showing that every r.e. set W is the solution set for a polynomial (with many variables), i.e. there is a polynomial $p(x, \bar{y})$ such that $W = \{x | \exists \bar{y}(p(x, \bar{y}) = 0\}$.

The language of arithmetic has symbols $+, \times, <, 0, 1, \varphi(\sigma, e, x, s)$. The Σ_n, Π_n formulas of arithmetic are defined as follows:

- $\Sigma_0 = \Pi_0$ are quantifier free formulas
- Σ_{n+1}: $\exists \bar{x}(F(\bar{x}))$ for $F \in \Pi_n$
- Π_{n+1}: $\forall \bar{x}(F(\bar{x}))$ for $F \in \Sigma_n$

An intermediate route puts bounded quantifiers into the language ($\exists x < s$, $\forall x < s$) as well as a few select predicates or functions β for coding finite sequences (of variable length) and the corresponding projection functions. (Explanation and/or thought exercise.) If we do so, $\Sigma_0 = \Pi_0$ have only bounded quantifiers. Note that the predicates defined by such formulas remain recursive.

Prenex normal form. Collapse like quantifiers. Move bounded quantifiers past unbounded ones.

A relation is Σ_n or Π_n if it is defined by a Σ_n or Π_n formula. A relation is in Δ_n if it is defined by both a Σ_n and a Π_n formula. Note that the notion of Δ_n is semantic rather than syntactic.

totp2 **Example 4.4.3** *Φ_e^f is total is a Π_2^f property of e (uniformly in f)*

lets3 **Example 4.4.4** *$\Phi_e^f \leq_T \Phi_i$ is a Σ_3^f relation of e and i (uniformly in f) as is $\Phi_e^f =_T \Phi_i$.*

Example 4.4.5 *$\Phi_e^f \wedge \Phi_i^f \equiv_T \Phi_j^f$ is a Π_4 relation of e, i and j (uniformly in f) but $\Phi_e^f \vee \Phi_i^f \equiv_T \Phi_j^f$ is a Σ_3 relation of e, i and j (uniformly in f). For the former simply write out the relation. For the latter seems to be essentially the same situation. However, atypically, here we can do better than the natural first attempt. Using the recursive function p defined in Example $\overset{\texttt{ujoin}}{2.1.10}$, we see that it is equivalent to $\Phi_{p(e,i)}^f \equiv_T \Phi_j^f$ and so, by the previous Example, Σ_3^f.*

Properties of $\Sigma_n, \Pi_n, \Delta_n$ Relations:

- If $A, B \in \Sigma_n$ then $A \cup B \in \Sigma_n$, $A \cap B \in \Sigma_n$, $\bar{A} \in \Pi_n$.
- If $A \in \Delta_n$ then $\bar{A} \in \Delta_n$.
- Σ_n is closed under projection. That is, if $A(x, y) \in \Sigma_n$ then $\{y : \exists x A(x, y)\} \in \Sigma_n$.
- Both Σ_n and Π_n are closed under bounded quantification. For $F \in \Sigma_n$,

  $$\exists x < sF \qquad \equiv \qquad \exists x\big(F(x) \wedge x < s\big),$$

  and

  $$\forall x < s \exists y_1 F \qquad \equiv \qquad \exists y\big(y \text{ is an } s\text{-tuple} \wedge \forall x < sF(x, \pi_x(y))\big).$$

  Note that this is sufficient because both checking tuple-hood and the projection functions are recursive so can use master function φ to represent them in our language.
- Uniformity.

We can relativize $\Pi_n^A, \Sigma_n^A, \Delta_n^A$ by adding a syntactic predicate $A(x)$ to the language and interpreting it in the semantics as the particular oracle set A. or multiple function parameters $\bar{h}$ .... see what need for defining forcing carry through for manipulations/normal forms. also hierarchy theorem could have $A, \bar{h}$

Proposition 4.4.6 *When we add in extra unary predicates or function symbols, the truth of Σ_0 formulas (even with bounded quantifiers) depends only on the values of the predicates (functions) below some value which can be computed recursively in the formula.*

We now see that we can define the recursive predicates as simply as possible.

Proposition 4.4.7 *$B \in \Sigma_1^A \Leftrightarrow B$ is r.e. in A.*

Move proof here.

So the recursive predicates (sets) in A are precisely the ones that are Δ_1^A.

Note that need bounded quantifiers for case with extra set or function symbols. Ask Frank Stephan for statement and proof. So do not have analog of Matijasevich et al. in relativized case.

4.5 The Hierarchy Theorem

hierarchy **Theorem 4.5.1 (Hiearchy Theorem)** *1. $B \in \Sigma^A_{n+1} \Leftrightarrow B$ is RE in some Π^A_n set.*

2. $A^{(n)}$ is Σ^A_n m-complete for $n > 0$.

3. $B \in \Sigma^A_{n+1} \Leftrightarrow B$ is RE in $A^{(n)}$.

4. $B \in \Delta^A_{n+1} \Leftrightarrow B \leq_T A^{(n)}$.

Proof. We need to use induction. Let us start with base case for (3), i.e. $B \in \Sigma^A_1 \Leftrightarrow B$ is RE in A. Suppose $x \in B \Leftrightarrow \exists y F(x, y, A)$ where F has only bounded quantifiers. Note that a formula which only contains bounded quantifiers is recursive in A. Let $\Phi^A_e(x) \downarrow \Leftrightarrow \exists y F(x, y, A)$ be the function which checks each value of y in turn and return " yes" answer if it finds one. So, $B = W^A_e$ and is RE in A. Conversely, suppose B is RE in A. Then $B = W^A_e$. This means that $x \in B \Leftrightarrow \exists\sigma\exists y\exists s\big(\varphi(\sigma, e, x, s) \wedge \sigma \subseteq A\big)$. ??Note that $\sigma \subseteq A$ is a bounded quantifier formula so we have a Σ^A_1 definition of B.??

To prove (1): The base case is $B \in \Sigma^A_1 \Leftrightarrow B$ is RE in some Π^A_0 set. Above we showed that if $B \in \Sigma^A_1$ then B is RE in A, which is Π^A_0. Conversely, if B is RE in some other Π^A_0 set, C, then since C is recursive in A, B is also RE in A so also use (3) to get $B \in \Sigma^A_1$.

For the induction, suppose $B \in \Sigma^A_{n+1}$. So $x \in B \Leftrightarrow \exists y F(x, y)$ where $F(x, y) \in \Pi^A_n$. In particular, B is $\Sigma^{F(x,y)}_1$ so is RE in $F(x, y)$ by the base case. Hence, B is RE in the Π^A_n set $F(x, y)$. Conversely, if B is RE in $Z \in \Pi^A_n$, by the base case, B is Σ^W_1. So, $x \in B \Leftrightarrow \exists\sigma, y, s\big(\varphi(\sigma, e, x, s) = y \wedge \sigma \subseteq Z\big)$ which is a Π^A_n definition.

To prove (2): We've previously shown that A' is the m-complete RE set. It remains to do the induction step. $A^{(n+1)} = \big(A^{(n)}\big)'$, which by the $n = 1$ case is the m-complete $\Sigma^{A^n}_1$ set. By induction, $A^n \in \Sigma^A_n$ so using (1) and that fact that being RE in X is the same as being RE in $\bar{X}$, we have that $A^{(n+1)} \in \Sigma^A_{n+1}$. For completeness, suppose $B \in \Sigma^A_{n+1}$. Then by (1), B is RE in some Π^A_n set C. So, B is RE in $\bar{C} \in \Sigma^A_n$. By the induction hypothesis, $A^{(n)}$ is Σ^A_n m-complete, so B is also RE in $A^{(n)}$. But X' is the 1-complete RE set, so $B \leq_m \big(A^{(n)}\big)' = A^{(n+1)}$.

To prove the induction step of (3): $B \in \Sigma^A_{n+1}$ if and only if B is RE in some Π^A_n set, C (by 1). This happens if and only if B is RE in $\bar{C} \in \Sigma^A_n$, which (by 2) happens if and only if B is RE in $A^{(n)}$.

For (4): $B \in \Delta^A_{n+1} \Leftrightarrow B \in \Sigma^A_{n+1} \cap \Pi^A_{n+1} \Leftrightarrow B$ is RE in $A^{(n)}$ and $\bar{B}$ is RE in $A^{(n)} \Leftrightarrow B \leq_T A^{(n)}$. ■

The hierarchy theorem tells us that one quantifier corresponds to one iteration of jump operator. For example, we have that if F is a predicate recursive in A, then $\exists x F \leq_T A'$ and $\exists x \forall y F \leq_T A''$.

Moreover, the hierarchy theorem also shows that the jump hierarchy is real: there are new sets at each levels. In particular, $A <_T A'$ implies that we have a strict hierarchy and $A^n \in \Sigma_n \setminus \Pi_n$. So we have $\Pi_n \neq \Sigma_n$ and

$$\Sigma_0 = \Pi_0 = \Delta_0 \subsetneq \Sigma_1 \subsetneq \Delta_1 \cdots$$

Diagram

Sets X definable in arithmetic: $X \leq_T 0^{(n)}$ some n

Relativize get arithmetic reducibility $X \leq_a Y$ and degrees corresponding to X definable from Y

4.5.1 Index sets

Define and samples

Exercise 4.5.2 *Prove that $\{e | W_e^A = \emptyset\}$ is 1-complete for Π_1^A.*

Exercise 4.5.3 *Prove that $\{e | W_e$ is infinite$\}$ is 1-complete for Π_2^A.*

Tot **Exercise 4.5.4** *Prove that $\{e | \Phi_e$ is total$\}$ is 1-complete for Π_2^A.*

Exercise 4.5.5 *Prove that $\{e | W_e$ is cofinite$\}$ is 1-complete for Σ_3^A.*

Exercise 4.5.6 *Prove that $\{e | W_e$ is recursive$\}$ is 1-complete for Σ_3^A. Hint: movable marker argument to fix location for diagonalization if not cofinite.*

jumphier

4.6 Jump Hierarchies

We would like a sense of what it means for a set to be computationally simple, or near 0 in degree.

Definition 4.6.1 *X is* low *if and only if $X' = 0'$.*

This is as close as you can get to measuring smallness using the jump. It says that the jump of X is as small (low) as possible. In many ways, such low sets look like recursive sets.

If we consider sets below $0'$, it is easy to see what it means for its jump to be as big as possible.

Definition 4.6.2 *For $X < 0'$: X is* high *if and only if $X' = 0''$.*

Again, many constructions which can be done below $0'$ can be done (more carefully) below any high set. Can we extend these notions of smallness and largeness beyond the degrees first jump?

Definition 4.6.3 *$X \in L_2$ if and only if $X'' = 0''$; for $X < 0'$, $X \in H_2$ if and only if $X'' = 0'''$.*
$X \in L_n$ if and only if $X^{(n)} = 0^{(n)}$; for $X < 0'$, $X \in H_n$ if and only if $X^{(n)} = 0^{(n+1)}$.

Now we generalize to degrees not necessarily below $0'$ again trying to capture the idea that the jump of a set is a small (low) or as large (high) as possible.

Definition 4.6.4 *$X \in GL_1$ if and only if $X' = X \vee 0'$; $X \in GH_2$ if and only if $X' = (X \vee 0')'$.*
$X \in GL_n$ if and only if $X^{(n+1)} = (X \vee 0')^{(n)}$; $X \in GH_n$ if and only if $X^{(n)} = (X \vee 0')^{(n)}$.

??mention future uses and connections as for domination, rates of growth as well as structural issues??

Chapter 5

embeddings

Embeddings into the Turing Degrees

5.1 Embedding Partial Orders in $\mathcal{D}$

So far, all we know about the structure of $\mathcal{D}$, the partial order of Turing degrees, is that it is an uppersemilattice with least element and the countable predecessor property. It also has an operator, the Turing jump, which is strictly increasing and closely related to the quantifier complexity of the definitions of sets and functions in arithmetic. The only specific degrees we know are $\mathbf{0}$ and the iterations of the jump beginning with $\mathbf{0}'$.

We now embark on our basic project of analyzing this structure beginning with the simplest algebraic or first order questions. Are there degrees other than $\mathbf{0}$ and the iterates of the jump operator? If so, where do they lie with respect to the ones we already know? Is $\mathcal{D}$ a linear order? If not, how "wide" is it? How far away from being a linear order? We start answering these questions by considering what is perhaps the simplest question and showing that $\mathcal{D}$ is not a linear order by constructing two Turing incomparable sets $A_0|_T A_1$ (and so degrees $\mathbf{a}_0|\mathbf{a}_1$).

KP **Theorem 5.1.1 (Kleene and Post)** $\exists A_0, A_1(A_0|_T A_1)$.

How can we approach such a result. We recast the desired properties of the sets we want to construct (Turing incomparability) as a list of simpler ones R_e called requirements. Then we choose an approximation procedure so that we can build a sequence of (pairs of) approximations $\alpha_{j,s}$ "converging" to A_j (for $j = 0, 1$) such that the information in one such approximation pair can be sufficient to guarantee that we satisfy one of the requirements in the sense that R_e is true of any pair of sets A_j with $A_j \supset \alpha_{j,s}$.

This basic approach, due to Kleene and Post, will be used for most of our constructions of degrees. It is a forerunner of the general method of forcing that we introduce in Chapter forcing 6

Our plan is to define A_0 and A_1 by constructing a sequence of approximations $\alpha_{j,s}$ which are finite binary strings (and so initial segments of characteristic functions). The intention is to make sure that $\alpha_{j,s} \subseteq \alpha_{j,s+1}$ for every s and, in the end, let $A_j = \cup_s \alpha_{j,s}$.

(Thus our notion of convergence is here quite simple: $A_j(x) = k \Leftrightarrow \exists s(\alpha_{j,s}(x) = k) \Leftrightarrow \exists s \forall t > s(\alpha_{j,t}(x) = k)$.)

The requirements necessary to guarantee that the sets so defined satisfy the theorem are:

$$R_{e,j} : \Phi_e^{A_j} \neq A_{1-j}$$

for all $e \in \mathbb{N}$, $j \in \{0, 1\}$. It is clear that if the sets we construct satisfy each requirement then the A_j are Turing incomparable and so satisfy the demands of the theorem.

What actions can we take to satisfy such a requirement? Given $\alpha_{j,s}$ ($j = 0, 1$), we want $\alpha_{j,s+1} \supseteq \alpha_{j,s}$ to guarantee that we satisfy $R_{e,j}$. For definiteness, let $j = 0$. We want $\alpha_0 \supseteq \alpha_{0,s}$, $\alpha_1 \supseteq \alpha_{1,s}$ such that for any $A_0 \supseteq \alpha_0$, $A_1 \supseteq \alpha_1$, $\Phi_e^{A_0} \neq A_1$. In other words,

$$\exists x \neg\big(\Phi_e^{A_0}(x) = A_1(x)\big).$$

Now any x will do here but for the sake of definiteness and simplicity of the construction, we choose x as the first place at which $\alpha_{1,s}$ is not defined (formally $x = \mathrm{dom}(\alpha_{1,s}) = |\alpha_{1,s}|$). We next try to satisfy the requirement with this x as the witness for the desired disagreement. We ask if $\exists \alpha_0 \supseteq \alpha_{0,s}\big(\Phi_e^{\alpha_0}(x)\downarrow\big)$. If so, we can choose any such α_0 as $\alpha_{0,s+1}$ and set $\alpha_{1,s+1} = \alpha_{1,s}\hat{}(1 - \Phi_e^{\alpha_0}(x))$ to diagonalize. Again for definiteness and simplicity we choose the *"least"* such α_0. To which ordering does "least" refer here? We use our master list of all (convergent) computations $\varphi(\sigma, e, x, t)$ from §codeTM 2.1, i.e. $\{\langle \sigma, e, x, t\rangle : \varphi(\sigma, e, x, t)\downarrow\}$. So "*least* α_0" refers to the σ in the least quadruple $\langle \sigma, e, x, s\rangle$ in this recursive set. From now on we, usually without comment, use "least" in this sense of being the first object enumerated by some given search procedure (that we know terminates).

By the standard properties of Turing machines given in §codeTM 2.1, if $A_0 \supseteq \alpha_0 = \alpha_{0,s+1}$ and $A_1 \supseteq \alpha_{1,s+1}$ then

$$\Phi_e^{A_0}(x) = \Phi_e^{\alpha_0}(x) \neq 1 - \Phi_e^{\alpha_0}(x) = A_1(x)$$

as desired.

What if no such α_0 exists? We *do nothing*, i.e. we set $\alpha_{i,s+1} = \alpha_{i,s}$ and hope for the best.

Indeed, a general principle of our constructions is we do the best we can, and if we cannot do anything useful, then we do nothing and hope for the best (i.e. that what we can do is enough). In this case, it *is* enough as we will see in the verification of the construction below. We now give the formal proof of Theorem KP 5.1.1 by describing the construction and the verification that it succeeds.

Proof. We begin with the construction.

Construction: We start with $\alpha_{j,0} = \emptyset$ for $j = 0, 1$ and proceed to inductively define $\alpha_{j,s}$ at stages s of the construction. So suppose we have defined $\alpha_{j,s}$. To define $\alpha_{j,s+1}$ consider $\langle e, i\rangle = s$ and act to satisfy requirement $R_{e,i}$ as follows. Let $x = |\alpha_{i,s}|$. If there is an $\alpha \supseteq \alpha_i, s$ such that $\Phi_e^\alpha(x)\downarrow$ let $\alpha_{i,s+1}$ be the least such α (as described above) and extend $\alpha_{1-i,s}$ to $\alpha_{1-i,s+1}$ by setting $\alpha_{1-i,s+1}(x) = 1 - \Phi_e^{\alpha_{i,s+1}}(x)$. If there is no such α, we let $\alpha_{j,s+1} = \alpha_j$ for $j = 0, 1$.

Verification: For any requirement $R_{e,j}$ consider stage $s = \langle e, j \rangle$ of the construction. If $\alpha_{1-j,s+1} \neq \alpha_{1-i,s}$, $\Phi_e^{\alpha_{j,s+1}}(x) \downarrow \neq \alpha_{1-j,s+1}(x)$ with $x = |\alpha_{1-j,s}|$. As $\alpha_{j,s+1} \subset A_j$ for $j = 0, 1$ by definition, the use property of computations (§codeTM 2.1) guarantees that $\Phi_e^{A_j}(x) \downarrow \neq A_{1-j}(x)$ and so we have satisfied $R_{e,j}$. If not, we claim that $\Phi_e^{A_j}(x) \uparrow$ and so $\Phi_e^{A_j}$ is certainly not A_{1-j} and we again satisfy $R_{e,j}$. The point here is that if $\Phi_e^{A_j}(x) \downarrow$ then, again by the basic properties of Turing computations, there is some finite initial segment $\hat{\alpha}$ of A_j such that $\Phi_e^{\hat{\alpha}}(x) \downarrow$. As $\alpha_{j,s} \subset A_j$, $\hat{\alpha}$ and $\alpha_{j,s}$ are compatible, their union α would satisfy the conditions forcing us to act to extend $\alpha_{1-j,s}$ at stage s of the construction contradicting our case assumption. Thus $\Phi_e^{A_i}(x) \uparrow$ and so we have satisfied $R_{e,j}$. ■

We next consider some questions about the construction and, in particular, about the its complexity and that of the sets it constructs.

nonterm **Question 5.1.2 (Nontermination)** *How do we know that this construction keeps going, i.e. that there is no point after which we always "do nothing".*

If the construction terminated in this way, then both A_0, A_1 would be finite, so certainly not Turing incomparable. So infinitely often we must extend the $\alpha_{j,s}$. We could include another set of requirements to guarantee this: $Q_{j,e} : |\alpha_{j,s}| \geq e$. These would, of course, be easy to satisfy and would make it both obvious and part of the formal verification that we extend the A_i infinitely often. However, we can see directly that this happens automatically: By the Padding Lemma (padding 2.1.12) there are infinitely many indices e such that $\Phi_e^A(x) = 0$ for every A and x and indeed one such that $\Phi_e(x)$ does not query its oracle at all for any x. At the stage at which we deal with the requirement $R_{e,j}$ for such an e, we automatically extend the approximation $\alpha_{1-j,s}$. Hence, both strings are extended infinitely often.

It is a common phenomenon that constructions in degree theory do more than one expects. We now see some other examples.

Question 5.1.3 (Complexity of the Construction) *The construction can be seen as simply the (production of the) double sequence $\alpha_{j,s}$. Of course, the sets A_j are recursive in the construction: By nontermination, there is, for any x, an s such that $\alpha_{j,s}(x)$ is defined for $j = 0, 1$. Thus, given the construction we can recursively find an s and k_j such that $\alpha_{j,s}(x) = k_j$. Then $A_j(x) = \alpha_{j,s}(x) = k_j$.*

As the sets A_j are Turing incomparable they are not recursive and neither then is the construction. How complicated is the construction and the sets A_0 and A_1? More precisely, can we find a bound such as $0^{(n)}$ on their degrees? Such a bound would also give definability properties such as A_j being Δ_{n+1} by the Hierarchy Theorem hierarchy 4.5.1.

We answer this question by showing that we may take $n = 1$. Let us look back at the construction. By recursion, we have $\alpha_{j,s}$ with $s = \langle e, i \rangle$. To calculate $\alpha_{j,s+1}$, we asked one question:

$$\exists \alpha \supseteq \alpha_{i,s} \big(\Phi_e^{\alpha}(x) \downarrow \big)?$$

This is a Σ_1 question so $0'$ can answer it and tell us which case to implement. In the second case of the construction, we recursively set $\alpha_{j,s+1} = \alpha_{j,s}$. In the first case of the construction, we can enumerate the master list $\{\langle\sigma, e, x, t\rangle : \varphi(\sigma, e, x, t)\downarrow\}$ and check recursively for an element with $\sigma \supseteq \alpha_{i,s}$. So, once $0'$ has told us which case we are in, everything else is recursive. Hence, the whole construction is recursive in $0'$ as are A_0 and A_1.

lowness **Question 5.1.4 (Complexity of the A_j)** *Where do A_0, A_1 lie in the jump hierarchy? Can we say more than just that they are recursive in $0'$. For example, are they low (or can we add something to the construction to make sure that they are low)? (Recall: A is low iff $A' \leq_T 0'$.)*

One approach to an answer to this question is to add new requirements:

$$N_{e,j} : \text{Make } \Phi_e^{A_j}(e)\downarrow \text{ if possible.}$$

We make a new list P_s of requirements including the old $R_{e,j}$ and the new $N_{e,j}$. At stages devoted to an $R_{e,j}$, i.e. $P_s = R_{e,j}$, we act as before. Suppose that at stage $s+1$ we are acting for $N_{e,j}$, i.e. $P_s = N_{e,j}$. We have $\alpha_{j,s}$ and ask if

$$\exists \alpha \supseteq \alpha_{j,s}\big(\Phi_e^{\alpha}(e)\downarrow\big)?$$

If the answer is yes, let $\alpha_{j,s+1}$ be the least such α and let $\alpha_{1,s+1} = \alpha_{1,s}$. On the other hand, if the answer is no, then do nothing and so let $\alpha_{j,s+1} = \alpha_{j,s}$ This is called *deciding (or forcing) the jump (of A_j at x).*

Claim 1: The construction is still recursive in $0'$: Our actions for requirements $R_{e,j}$ are the same as before. For $N_{e,j}$, $0'$ can decide if $\exists\alpha \supseteq \alpha_{j,s}\big(\Phi_e^{\alpha}(e)\downarrow\big)$. If there is such an α, we can find the least one recursively and compute $\alpha_{j,s+1}$. If not, there is nothing to do and $\alpha_{j,s} = \alpha_{j,s+1}$.

Claim 2: We can compute A_j' from $0'$. Since the whole construction is recursive in $0'$, $0'$ can go along the construction until it gets to the stage s at which we act for $N_{e,j}$. If $\Phi_e^{\alpha_{j,s+1}}(e)\downarrow$, then clearly $\Phi_e^{A_j}(e)\downarrow$, i.e. $e \in A_j'$. If not then no extension of α of $\alpha_{j,s}$ makes $\Phi_e^{\alpha}(e)\downarrow$ and so (as in the verification), $\Phi_e^{A_j}(e)\uparrow$, i.e. $e \notin A_j'$.

As was the case for Question nonterm 5.1.2, more happens in our original construction than is evident. It is possible to prove that the A_j constructed by the original construction (for Theorem KP 5.1.1) are already low (Exercise autolow 5.1.8).

Question 5.1.5 (Relativization) *What about incomparable degrees above any given* $\mathbf{x}$*?*

This is just an exercise in relativization. ??Ref to earlier By relativizing, we mean that at each part of the construction where we have as an oracle α_j, we instead use $X \oplus \alpha_j$ as the oracle . At the end, we build $X \oplus A_j$.??

relKP **Exercise 5.1.6** *Prove that for every set X there are A_j, $j = 0, 1$, such that $X \oplus A_0 |_T X \oplus A_1$ and $(X \oplus A_j)' \equiv_T X'$ for $j = 0, 1$.*

Question 5.1.7 (A Generalization) *Can we build more than two incomparables?*

We can easily change the list of requirements to

$$P_{e,j,k} : \text{If } j \neq k,\ \Phi_e^{A_j} \neq A_k.$$

for $e, j, k \in \mathbb{N}$ and handle them as we did the $R_{e,j}$ with $j = 0, 1$. Thus, we can produce countably many low pairwise incomparables between 0 and $0'$, indeed all with jumps uniformly recursive in $\mathbf{0}'$ (Exercise lowincomp 5.1.12).

autolow **Exercise 5.1.8** *Show that the sets A_i of the original construction (for Theorem KP 5.1.1) are already low.*

Exercise 5.1.9 *Add requirements to the construction of Theorem KP 5.1.1 to make the join of the sets constructed low, i.e. $(A_0 \oplus A_1) \equiv_T 0'$.*

We can strengthen the notion of lowness and prove a bit more:

slow **Definition 5.1.10** *A is* superlow *if $A' \leq_{tt} 0'$.*

Exercise 5.1.11 *Prove that the sets constructed in Theorem KP 5.1.1 are superlow.*

lowincomp **Exercise 5.1.12** *Prove that there are sets $A_i \leq_T 0'$, for $i \in \mathbb{N}$, such that $A_i | A_j$ for $i \neq j$. Moreover all these sets can made be low as well.*

colnot **Notation 5.1.13** *??(earlier) Given any sequence $\langle A_i | i \in I\rangle$ of sets we let $\oplus\{A_i | i \in I\} = \{\langle i, x\rangle \,| i \in I \;\&\; x \in A_i\}$. Conversely, given any set A we let $A^{[i]}$ denote the set $\{x | \langle i, x\rangle \in A\}$. We let $A^{[\hat{i}]} = \oplus\{A_j | i \neq j\} = \{\langle j, x\rangle \,| i \neq j \;\&\; x \in A_j\}$. We use the same notation for binary strings σ: $\sigma^{[i]}(x) = \sigma(\langle i, x\rangle)$ and $\sigma^{[\hat{i}]}(\langle j, x\rangle) = \sigma(\langle j, x\rangle)$ for $j \neq i$ and $\sigma^{[\hat{i}]}(\langle i, x\rangle) = 0$ for $\langle i, x\rangle \in \operatorname{dom}\sigma$.?? Warning: changed definition of $A^{[i]}$ so that $\oplus A^{[i]} = A$. Also made $\sigma^{[i]}$ and $\sigma^{[\hat{i}]}$ into binary strings??*

As a significant generalization of Theorem KP 5.1.1 and even of Exercise lowincomp 5.1.12, we can try to embed arbitrary countable partial orders $\mathcal{P}$ in $\mathcal{D}$ or in $\mathcal{D}(\leq 0')$ or in the low degrees.

Consider any countable partial order $\mathcal{P}$ with domain $\{p_0, p_1, \ldots\}$ and partial order $\leq_{\mathcal{P}}$. We want to construct A_i such that $A_i \leq_T A_j$ if and only if $p_i \leq_{\mathcal{P}} p_j$. To do so, we build auxiliary sets C_i and, in an attempt to mimic the order $\leq_{\mathcal{P}}$, we let $A_j = \oplus\{C_i : p_i \leq_{\mathcal{P}} p_j\}$.

The first question is does this succeed to the extent that $p_i \leq_{\mathcal{P}} p_j$ implies that $A_i \leq_T A_j$? Well, $\langle k, x\rangle \in A_i \Leftrightarrow x \in C_k \wedge p_k \leq_{\mathcal{P}} p_i$ by definition but, as $p_i \leq_{\mathcal{P}} p_j$ (and the ordering is transitive), this is the same (again by definition) as $p_k \leq_{\mathcal{P}} p_i \;\wedge\; \langle k, x\rangle \in A_j$. So if $\leq_{\mathcal{P}}$ is recursive, $i \leq_{\mathcal{P}} j$ implies that $A_i \leq_T A_j$. We can use this fact to embed recursive partial orders in the low degrees by using the constructions above to guarantee incomparability when needed and this simple argument for recursive $\mathcal{P}$ to guarantee comparability when needed. If a partial order is not recursive, it is at least recursive in some oracle so relativizing the proof for recursive partial orders gives an embedding into $\mathcal{D}$. Perhaps this is the best we can do – it is not intuitively obvious or even, perhaps even plausible that $\mathcal{D}(\leq 0')$ is a universal countable partial order, i.e. every countable partial order can be embedded in it. (See Appendix ??.)

We begin our proof that it is by constructing a recursive universal partial order. The construction is an example of the method of finite approximations being used to build sets with properties not necessarily expressed in terms of Turing degrees. The idea is related to the proof that $\mathbb{Q}$ is a countable universal linear order (Appendix ??). We then embed it into $\mathcal{D}(\leq \mathbf{0}')$ using the methods of Theorem $\overset{\texttt{KP}}{5.1.1}$. We start with a Lemma that will provide the basic steps of our construction.

1ext **Lemma 5.1.14** *Given a partial order $\mathcal{P}$, a suborder $\mathcal{R}$ and an extension of $\mathcal{R}$ to a partial order $\hat{\mathcal{R}}$ containing exactly one new element z (not in $\mathcal{P}$), there is an extension $\hat{\mathcal{P}}$ of $\mathcal{P}$ containing z as its only new element that also extends $\hat{\mathcal{R}}$.*

Proof. We let $\hat{P} = P \cup \{z\}$. To define the desired ordering $\leq_{\hat{\mathcal{P}}}$ on $\hat{P}$, we must specify when $p \leq_{\hat{\mathcal{P}}} z$ and when $z \leq_{\hat{\mathcal{P}}} p$ for $p \in P - \hat{R}$ and verify that we have defined a partial order. We set $p \leq_{\hat{\mathcal{P}}} z \Leftrightarrow (\exists r \in R)(p \leq_{\mathcal{P}} r \ \&\ r \leq_{\hat{\mathcal{R}}} z)$. Similarly, we set $z \leq_{\hat{\mathcal{P}}} p \Leftrightarrow (\exists r \in R)(r \leq_{\mathcal{P}} p \ \&\ z \leq_{\hat{\mathcal{R}}} r)$. All other instances of order relations are as in $\mathcal{P}$ and $\hat{\mathcal{R}}$.

Clearly $\leq_{\hat{\mathcal{P}}}$ is reflexive and extends both $\mathcal{P}$ and $\hat{\mathcal{R}}$. We must check that it is transitive. Consider any $u \leq_{\hat{\mathcal{P}}} v$ and $v \leq_{\hat{\mathcal{P}}} w$. We must show that $u \leq_{\hat{\mathcal{P}}} w$. Certainly if none of u, v or w is z (so all are in P) there is nothing to prove as $\leq_{\mathcal{P}}$ is transitive. If two of them are z, the desired conclusion is again immediate. If only $w = z$ then, by definition, there is an $r \in R$ such that $v \leq_{\mathcal{P}} r \leq_{\hat{\mathcal{R}}} z$. As, in this case, $u \leq_{\mathcal{P}} v \leq_{\mathcal{P}} r$ and $\leq_{\mathcal{P}}$ is transitive, $u \leq_{\hat{\mathcal{P}}} z$ by definition. Similarly, if only $u = z$ then, by definition, there is an $r \in R$ such that $z \leq_{\hat{\mathcal{R}}} r \leq_{\mathcal{P}} v$. As, $v \leq_{\mathcal{P}} w$ (and $\leq_{\mathcal{P}}$ is transitive), $z \leq_{\hat{\mathcal{P}}} w$ by definition. Finally, if only $v = z$ then there are r_1 and r_2 in R such that $u \leq_{\mathcal{P}} r_1 \leq_{\hat{\mathcal{R}}} z$ and $z \leq_{\hat{\mathcal{R}}} r_2 \leq_{\mathcal{P}} w$. As $\leq_{\hat{\mathcal{R}}}$ is transitive, $r_1 \leq_{\hat{\mathcal{R}}} r_2$ but as both are in $\mathcal{R}$ which is a suborder of $\mathcal{P}$, $u \leq_{\mathcal{P}} r_1 \leq_{\mathcal{P}} r_2 \leq_{\mathcal{P}} w$ and so $u \leq_{\mathcal{P}} w$ and $u \leq_{\hat{\mathcal{P}}} w$ as required. ■

univpo **Theorem 5.1.15** *There is a recursive universal countable partial order $\mathcal{Q}$, i.e. a recursive partial order $\mathcal{Q}$ such that every countable partial order $\mathcal{P}$ can be embedded in $\mathcal{Q}$.*

Proof. We build $\mathcal{Q}$ by finite approximations, $\mathcal{Q} = \cup \mathcal{Q}_s$. At stage $s + 1$ we have a finite partial order $\mathcal{Q}_s$ and extend it to $\mathcal{Q}_{s+1}$ such that for every suborder $\mathcal{M}$ of $\mathcal{Q}_s$, every one element partial order extension $\hat{\mathcal{M}}$ of $\mathcal{M}$ is realized in $\mathcal{Q}_{s+1}$. That is, for every subset $\mathcal{M} \subset \mathcal{Q}_s$, and extension $\hat{\mathcal{M}}$ of $\mathcal{M}$ with $\hat{M} = M \cup \{z\}$ for some $z \notin Q_s$ there is an embedding of $\hat{\mathcal{M}}$ into $\mathcal{Q}_{s+1}$ which is the identity on $\mathcal{M}$. To do this we list all the suborders $\mathcal{M}_j$ of $\mathcal{Q}_s$ and all of their one element extensions $\mathcal{M}_{j,k}$ and apply Lemma $\overset{\texttt{1ext}}{5.1.14}$

successively to each $\langle j,k\rangle$ to produce a sequence of partial orders $\mathcal{Q}_{s,\langle j,k\rangle}$ each extending the previous one and an embedding of $\mathcal{M}_{j,k}$ into $\mathcal{Q}_{s,\langle j,k\rangle}$ which is the identity on $\mathcal{M}_j$. Clearly, this is a recursive procedure and the final partial order so produced is the desired Q_{s+1}.

To see that $\mathcal{Q} = \cup\, \mathcal{Q}_s$ is universal, consider any countable partial order $\mathcal{P}$ with $P = \{p_0, p_1, \ldots\}$. We define the embedding $f : \mathcal{P} \to \mathcal{Q}$ by recursion. Start with $f(p_0) = q_0$. Given an embedding f_n of $\mathcal{P} \restriction \{p_i | i < n\}$ into $\mathcal{Q}$, choose s such that the range of this finite embedding is contained in $\mathcal{Q}_s$. This range is then a suborder $\mathcal{M}$ of $\mathcal{Q}_s$. Consider the one element extension $\hat{\mathcal{M}}$ of $\mathcal{M}$ that is isomorphic to $\mathcal{P} \restriction \{p_i | i < n+1\}$. By our construction of $\mathcal{Q}_{s+1}$ there is an extension of f_n to an embedding of $\mathcal{P} \restriction \{p_i | i < n+1\}$ into $\mathcal{Q}_{s+1}$ which agrees with f_n on $\{p_i | i < n\}$. This map is the desired f_{n+1} and $f = \cup f_n$ is the desired embedding of $\mathcal{P}$ into $\mathcal{Q}$. ■

embrecpo **Proposition 5.1.16** *Every recursive partial order $\mathcal{P} = (P, \leq_{\mathcal{P}})$ can be embedded in $\mathcal{D}$.*

Proof. Let $P = \{p_i | i \in \mathbb{N}\}$. We build sets C_i and let $A_i = \oplus_j\{C_j : p_j \leq_{\mathcal{P}} p_i\}$. As argued above, if $p_k \leq_{\mathcal{P}} p_j$ then $A_k \leq_T A_j$ since $\leq_P$ is recursive. We now turn to the requirements needed to guarantee that if $p_k \not\leq_{\mathcal{P}} p_j$ then $A_k \not\leq_T A_j$:

$$R_{k,j,e} : \text{ If } p_k \not\leq_{\mathcal{P}} p_j \text{ then } \Phi_e^{A_j} \neq A_k.$$

As for our approximations, at every stage s of our construction we will have defined a finite set Γ_s of finite binary strings $\gamma_{j,s}$. At the end, we set $C_j = \cup \gamma_{j,s}$. Given any finite set Γ of finite binary strings γ_j (such as the Γ_s), the corresponding approximation for the A_i is given by

$$A_i[\Gamma] = \oplus_{j \in \Gamma}\{\gamma_j : p_j \leq_{\mathcal{P}} p_i\}$$

i.e. for $p_j \leq_{\mathcal{P}} p_i$, $A_i[\Gamma]$ is defined at $\langle j,x\rangle$ if and only if $\gamma_j \in \Gamma$ and $\gamma_j(x)$ is defined in which case $A_i[\Gamma](\langle j,x\rangle) = \gamma_j(x)$. If $p_j \not\leq_{\mathcal{P}} p_i$, we let $A_i[\Gamma](\langle j,x\rangle) = 0$. We write $A_{i,s}$ for $A_i[\Gamma_s]$ and so $A_i = \cup A_{i,s}$.

Construction: We begin with $\Gamma_0 = \emptyset$. At stage $s+1$ with $s = \langle k,j,e\rangle$, we have $A_{j,s}$ and $A_{k,s}$ the partial characteristic functions determined by the $\gamma_{i,s}$ so far defined (Γ_s) and we wish to act for $R_{k,j,e}$. To guarantee that in the end $\Phi_e^{A_j} \neq A_k$, we could try to take (as in Theorem $\overset{\text{KP}}{5.1.1}$) $x = |\gamma_{k,s}|$ and ask if there is extension Γ of Γ_s such that $\Phi_e^{A_j[\Gamma]}(x)\downarrow$ to diagonalize. The problem is that an extension of Γ_s which guarantees convergence might also determine the value $A_k(x)$, so we might not be able to diagonalize.

To make sure x does not interfere with the computation from $A_j[\Gamma]$, we want an $x = \langle n,y\rangle$ such that $p_n \not\leq_{\mathcal{P}} p_j$. Also, to be able to define A_k at x however we want so as to be able to diagonalize, we need $p_n \leq p_k$ (otherwise the relevant column is always empty). We also need $\langle n,y\rangle \geq |\gamma_{k,s}|$. So we want $p_n \not\leq p_j$ and $p_n \leq_{\mathcal{P}} p_k$. By assumption, $p_k \not\leq_{\mathcal{P}} p_j$, so we choose $n = k$ and let $x = \langle k, |\gamma_{k,s}|\rangle$.

Now, ask if there is an extension Γ of Γ_s such that $\Phi_e^{A_j[\Gamma]}(x)\downarrow$. If so, it is clear from the definition of $A_j[\Gamma]$ that this computation only depends on $\gamma_i \in \Gamma$ for $p_i \leq_{\mathcal{P}} p_j$ and so (as $p_k \not\leq_T p_j$) we may assume that if $\gamma_k \in \Gamma$ then it is undefined at x. If such an extension exists choose the least one $\hat{\Gamma}$ and let Γ_{s+1} extend $\hat{\Gamma}$ so that $\gamma_{k,s+1} \in \Gamma_{s+1}$ and $\gamma_{k,s+1}(x) = 1 - \Phi_e^{A_j[\Gamma]}(x) = 1 - \Phi_e^{A_j,s+1}(x)$. If there is no such extension, set $\Gamma_{s+1} = \Gamma_s$.

Verification: To see that the construction satisfies each requirement $R_{k,j,e}$ we may assume that $p_k \not\leq_{\mathcal{P}} p_j$ and consider the stage $s = \langle k, j, e\rangle$ and the corresponding x. If $\Phi_e^{A_j,s+1}(x)\downarrow$ then it equals $\Phi_e^{A_j}(x)$ and is different from $A_k(x)$ as required. If $\Phi_e^{A_j,s+1}(x)\uparrow$ then no extension Γ of Γ_s makes $\Phi_e^{A_j[\Gamma]}(x)\downarrow$. On the other hand, if $\Phi_e^{A_j}(x)\downarrow$ then the finite amount of information about A_j needed to produce this convergence provides a Γ such that $\Phi_e^{A_j[\Gamma]}(x)\downarrow$ and $A_j[\Gamma] \subset A_j$. By the definition of A_j this Γ extends Γ_s for the desired contradiction. ■

Exercise 5.1.17 *Do we need to worry that the $\cup\gamma_{j,s}$ in the construction for Proposition* embrecpo *5.1.16 might not be a total function? What changes in the construction would make this obvious? Why are none needed?*

embpo **Corollary 5.1.18** *Every countable partial order $\mathcal{P}$ can be embedded in $\mathcal{D}$.*

Proof. Let $\mathcal{Q}$ be the recursive universal countable partial order constructed in Theorem univpo 5.1.15 and f its embedding into $\mathcal{D}$ as given by Proposition embrecpo 5.1.16.As $\mathcal{Q}$ is universal there is an embedding $g{:}\mathcal{P} \to \mathcal{Q}$ and then $g \circ f$ is the desired embedding of $\mathcal{P}$ into $\mathcal{D}$. ■

1qtth **Corollary 5.1.19** *The one-quantifier theory of $(\mathcal{D}, \leq_T)$ is decidable.*

Proof. A one-quantifier existential sentence φ begins with a string $\exists x_0 \exists x_1 \cdots \exists x_n$ of existential quantifiers and is followed by a quantifier free matrix built from atomic formulas of the form $x_i \leq x_j$ or $x_i = x_j$ for $i, j \leq n$. Note that if we can decide whether an existential sentence is true or false in $\mathcal{D}$ then we can flip the answers to decide if universal sentences are true or false. We claim that $\mathcal{D} \vDash \varphi$ if and only if there is a partial order of size $n+1$ which satisfies φ and that this question can be answered recursively.

First note that for any partial order $\mathcal{P}$, $\mathcal{P} \vDash \varphi$ if and only if some subpartial order $\mathcal{Q}$ of $\mathcal{P}$ of size at most $n+1$ satisfies φ. The point here is that the truth of atomic statements about elements of $\mathcal{Q}$ are the same in $\mathcal{P}$ and $\mathcal{Q}$. So satisfaction in $\mathcal{Q}$ implies satisfaction (via the same witnesses) in $\mathcal{P}$. In the other direction, if φ is true in $\mathcal{P}$, then the suborder $\mathcal{Q}$ consisting of the witness in $\mathcal{P}$ needed to verify φ form a subpartial order $\mathcal{Q}$ of size at most $n+1$. As we know from Proposition embrecpo 5.1.16 that every finite partial order is isomorphic to a subpartial order of $\mathcal{D}$, $\mathcal{D} \vDash \varphi$ if and only if some finite partial order $\mathcal{Q}$ of size at most $n+1$ satisfies φ. We can decide this last condition recursively by listing all the (finitely many) partial orders $\mathcal{Q}_k$ of size at most $n+1$ and then checking for each $\mathcal{Q}_k$ if it satisfies φ by checking all the (finitely many) possible instantiations of the x_i for $i \leq n$ as elements of $\mathcal{Q}_k$. ■

Exercise 5.1.20 *If the recursive partial order $\mathcal{P}$ of Proposition* embrecpo *5.1.16 has a least element 0, then embedding f into $\mathcal{D}$ can be chosen such that $f(0) = \mathbf{0}$. Corollary* embpo *5.1.18 can then be extended to partial orders with least element and Corollary* 1qtth *5.1.19 to the language with a constant for its least element* **0**.

We ask the following questions about the proof Proposition embrecpo *5.1.16:*

Question 5.1.21 *How complicated are the images of the partial order under the embedding?*

We claim that $A_i \leq_T 0'$ uniformly. Indeed the whole construction and so the C_i are (uniformly) recursive in $0'$. To compute $A_i(x)$ where $x = \langle j, n \rangle$ we first ask if $p_j \leq p_i$ (the partial ordering is recursive). If not, $A_i(x) = 0$. If so, we can follow the construction recursively in $0'$ until it is decided if $x \in C_j$.

Question 5.1.22 *Can we ensure that all the A_i are low?*

We can add requirements

$$N_e : \text{ Make } \Phi_e^{\oplus A_i}(e) \downarrow \text{ if we can.}$$

To act for N_e still takes just a $0'$ question.

emb<0' **Corollary 5.1.23** *Every countable partial older can be embedded in $\mathcal{D}(\leq \mathbf{0}')$ and so its one quantifier theory is decidable.*

Exercise 5.1.24 *Make the constructions suggested here precise, verify the associated assertions and prove Corollary* emb<0' *5.1.23.*

An alternative approach to these results begins with strengthened versions of incomparability.

Definition 5.1.25 *The set $\{A_i : i \in \mathbb{N}\}$ is* independent *if no A_i is computable from the join of finitely many of the other A_j. The set $\{A_i : i \in \mathbb{N}\}$ is* very independent *if $A_i \not\leq_T \oplus_{j \neq i} A_j$ for all i.*

Very independent implies independent because $A_{i_1} \oplus \cdots \oplus A_{i_n} \leq_T \oplus_{j \neq i} A_j$ if no $i_k = i$: $x \in A_i \Leftrightarrow \langle i, x \rangle \in \oplus_{j \neq i} A_j$. However, while independence is a degree theoretic notion, very independence is not. This is proved in the following exercises.

Exercise 5.1.26 *Prove that there is a countable set $\{A_i : i \in \mathbb{N}\}$ which is very independent. Indeed, one can make $\oplus A_i$ low.*

Exercise 5.1.27 *Construct $\{A_i : i \in \mathbb{N}\}, \{B_i : i \in \mathbb{N}\}$ such that $\{A_i : i \in \mathbb{N}\}$ is very independent, $\{A_i : i \in \mathbb{N}\}$ is not, but $A_i \equiv_T B_i$.*

Definition 5.1.28 *An* uppersemilattice (usl) *is a partially ordered set $\mathcal{P}$ such that every pair of elements x, y in $\mathcal{P}$, has a least upper bound, $x \vee y$.??*

??Some of these to Appendix and just cite here??

Exercise 5.1.29 *Every usl $\mathcal{L}$ is locally countable, ??i.e. for any finite $F \subset L$ the subusl $\mathcal{F}$ of $\mathcal{L}$ generated by F (i.e. the smallest one containing F) is finite??. Moreover, there is a uniform recursive bound on $|\mathcal{F}|$ that depends only on $|F|$.*

Exercise 5.1.30 *Given finite usls $Q \subset P$ and an usl extension $\hat{Q}$ of Q generated over Q by one new element (with $\hat{Q} \cap P = Q$), prove that there is an usl extension $\hat{P}$ of P containing $\hat{Q}$.*

Exercise 5.1.31 *Prove that there is a recursive usl $\mathcal{L}$ such that every countable usl can be embedded in it (as an usl).*

Exercise 5.1.32 *Every countable usl $\mathcal{L}$ can be embedded in $\mathcal{D}$ and even in $\mathcal{D}(\leq \mathbf{0}')$ (preserving $\vee$ as well as $\leq$ and 0 if $\mathcal{L}$ has a least element). Hint: Use a very independent set C_i. If $\mathcal{L} = \{l_i\}$ send l_i to $\oplus\{C_j | l_j \not\geq l_i\}$.*

Exercise 5.1.33 *The one quantifier theory of $\mathcal{D}$ as an upper semilattice is decidable.*

Refs to Appendix??

Notes: The finite extension method for constructing degrees was developed in Kleene and Post [1954]. This paper was the seminal paper on the structure of the Turing degrees. Kleene and Post proved, among others, Theorem KP 5.1.1, the existence of countable independent sets, and Proposition embrecpo 5.1.16 for finite partial orders and that all these theorems are true in the degrees below $\mathbf{0}'$. Sacks [1961] and [1963] contain Corollary embpo 5.1.18 and much more. Corollary 1qtth 5.1.19 is pointed out in Lerman [1972].

We will see in Theorem latemb 6.3.1 that every countable lattice can be embedded in $\mathcal{D}$ but not by the methods used here in the sense that there is no countable lattice $\mathcal{L}$ which is countably universal, let alone a recursive one. Indeed local finiteness fails and there are $2^{\aleph_0}$ many lattices generated by four elements ??ref??. We provide such with seven generators in §ESS 6.4. ??Appendix??

What about uncountable partial orders, usls and lattices? Of course, to be embeddable in $\mathcal{D}$ they must have the countable predecessor property, i.e. $\{y | y \leq x\}$ is countable for every x. Sacks [1961] shows that all partial orders of size $\aleph_1$ with the countable predecessor property can be embedded into $\mathcal{D}$. For uppersemilattices this follows from Abraham and Shore [1986] where the embedding is made onto an initial segment of $\mathcal{D}$. (Initial segments of $\mathcal{D}$ are considered in Chapters MinDegs ?? and initialseg 10.) Some simple examples of suborderings of $\mathcal{D}$ of size $2^{\aleph_0}$are provided in §Treec 5.4. Sacks [1961] shows that all those with the countable successor property can be embedded. However, it is consistent that $2^{\aleph_0} = \aleph_2$ and there is an usl of size $\aleph_2$ with the countable predecessor property which cannot be embedded in $\mathcal{D}$ (Groszek and Slaman [1983]. It is a long standing open question if every partial order of size $2^{\aleph_0}$ with the countable predecessor property can be embedded in $\mathcal{D}$ (Sacks [1963]).

5.2 Extensions of embeddings

The next step after embedding results are what are called extension of embedding results. For example, we consider an arbitrary degree $\mathbf{x}$ and we ask if there is always a $\mathbf{y}$ such that $\mathbf{y} <_\mathbf{T} \mathbf{x}$ or $\mathbf{y} >_\mathbf{T} \mathbf{x}$ or $\mathbf{y}|_T\mathbf{x}$. To make this question nontrivial we want to require that $\mathbf{x} > \mathbf{0}$. While we could add a constant for $\mathbf{0}$ to our language, this is not necessary as long as we move to the general case of extension of embeddings questions. Here we consider an arbitrary sequence $\bar{\mathbf{x}} = \langle \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\rangle$ of degrees with some specified order structure $\mathcal{P}$ and we ask if there is always another sequence $\bar{\mathbf{y}}$ such that $\langle \bar{\mathbf{x}}, \bar{\mathbf{y}}\rangle$ satisfies some given extension $\mathcal{Q}$ of $\mathcal{P}$. Thus the general questions of this type are of the form $\forall \bar{x}(\Theta(\bar{x}) \rightarrow \exists \bar{y}\Phi(\bar{x}, \bar{y}))$ where Θ and Φ are quantifier free. With this generality, we can rephrase, for example, the question of whether for every $\mathbf{x} > \mathbf{0}$ there is a $\mathbf{y}|\mathbf{x}$ as whether $(\forall \mathbf{x}_0, \mathbf{x}_1)(\mathbf{x}_1 < \mathbf{x}_0 \rightarrow (\exists \mathbf{y}_0)(\mathbf{y}_0|\mathbf{x}_0))$. Indeed, for any such sentence about $\mathcal{D}$ with a constant symbol for $\mathbf{0}$, we can find an equivalent sentence of the same form in the language with just $\leq$ (Exercise elim0 5.2.1). Another basic example is the question of whether $\mathcal{D}$ is dense, i.e. if for every $\mathbf{x}_0 < \mathbf{x}_1$ there is a $\mathbf{y}$ such that $\mathbf{x}_0 < \mathbf{y} < \mathbf{x}_1$. Here the answer is "no" but more complicated techniques are needed to prove it. Indeed by Theorem spmindeg 9.2.21, there are minimal degrees $\mathbf{x}$, i.e. $\mathbf{0} < \mathbf{x}$ but with no $\mathbf{y}$ such that $\mathbf{0} < \mathbf{y} < \mathbf{x}$.) Such questions are special cases of the decision problem for all two quantifier sentences in $\mathcal{D}$. We will eventually see that the answers to these extension of embedding questions are enough to decide the full 2-quantifier theory of $\mathcal{D}$ (2qtth 10.4).

elim0 **Exercise 5.2.1** *Consider the theory of partial orders with least element in the language* $(\leq, 0)$ *where we add on the axiom* $\forall z(0 \leq z)$ *to guarantee that the constant* 0 *always is interpreted as the least element of the partial order. Show that for any* 2*-quantifier sentence* $\forall\bar{x}\exists\bar{y}\Psi(\bar{x},\bar{y})$ *in this language there is an equivalent* 2*-quantifier sentence without the constant symbol. Also show that if the original sentence is of the form* $\forall \bar{x}(\Theta(\bar{x}) \rightarrow \exists \bar{y}\Phi(\bar{x}, \bar{y}))$ *then the equivalent sentence without* 0 *can also be taken to be of this form. Hint: Extend the lists of variables* $\bar{x}$ *and* $\bar{y}$ *each by one new one* v *and* w*, respectively. Then consider the sentence* $(\forall \bar{x}, v)(\exists \bar{y}, w)(w \not\leq v \vee \hat{\Phi}(\bar{x},\bar{y}))$ *where* $\hat{\Phi}$ *is gotten from* Φ *by replacing* 0 *by* v*.*

We begin with a simple but basic example mentioned above.

coneavoid **Theorem 5.2.2 (Avoiding cones)** *For every* $A > 0$ *there is* B *such that* $A|_TB$*. Indeed, there is such a* $B \leq_T A'$*.*

Proof. Given a set A, we build B such that $A \not\leq_T B$, $B \not\leq_T A$. There are two kinds of requirements:

$$P_e : \Phi_e^A \neq B \qquad\qquad Q_e : \Phi_e^B \neq A.$$

The construction uses finite binary string approximations β_s to B. At the end, we let $B = \cup_s\beta_s$. We order these requirements in an arbitrary way as R_s.

Construction: We begin with $\beta_0 = \emptyset$. At stage $s+1$ we have β_s and work to satisfy R_s by constructing an appropriate β_{s+1}. If $R_s = P_e$, we ask if $\Phi_e^A(|\beta_s|)\uparrow$. If so, then P_e is satisfied so do nothing, i.e. $\beta_{s+1} = \beta_s$. Otherwise, we let $\beta_{s+1} = \beta_s\hat{\ }(1-\Phi_e^A(|\beta_s|))$. So, $B(|\beta_s|) = \beta_{s+1}(|\beta_s|) \neq \Phi_e^A(|\beta_s|)$ and again the requirement is satisfied. Observe that at this stage we ask a question that A' can answer and then carry out a procedure recursive in A.

If $R_s = Q_e$, we ask if there is an x and an extension σ of β_s such that $\Phi_e^\sigma(x)\downarrow\neq A(x)$. If no such extension exists, do nothing. If there is such an extension, let β_{s+1} be the least such extension. Observe that we here asked a Σ_1^A question followed by a recursive procedure based on the answer, so this step is also recursive in A'.

Verification: We have already noted that all the P_e are satisfied and that the construction, $\langle\beta_s\rangle$, and so B is recursive in A'. Suppose we fail to satisfy Q_e. Then at the stage s with $R_s = Q_e$ there was no x and $\sigma \supset \beta_s$ such that $\Phi_e^\sigma(x)\downarrow\neq A(x)$. If $\Phi_e^B(x)\uparrow$ for any x then Q_e is satisfied. Otherwise, we claim that A is recursive: To compute $A(x)$, look for any $\sigma \supseteq \beta_s$ such that $\Phi_e^\sigma(x)\downarrow$. There is one since $\Phi_e^B(x)\downarrow$ and $B \supset \beta_s$. By our case assumption, the value computed with oracle σ must be $A(x)$. Thus we have recursively computed A for a contradiction and so Q_e is satisfied. ■

coneavoid' **Exercise 5.2.3** *Modify the construction of Theorem* coneavoid 5.2.2 *to make $B' \leq_T A'$. Alternatively show that the construction already guarantees that $B' \leq_T A'$.*

unctblch **Exercise 5.2.4** *Every maximal chain (i.e. linearly ordered subset) in $\mathcal{D}$ is uncountable.*

ntconeavoid **Exercise 5.2.5** *For every countable set of nonrecursive degrees there is a degree incomparable with each of them.*

axantichain **Exercise 5.2.6** *Every maximal antichain (i.e. set of pairwise incomparables) in $\mathcal{D}$ other than $\{\mathbf{0}\}$ is uncountable.*

unctindep **Exercise 5.2.7** *Every maximal independent set of degrees is uncountable.*

We now turn to a weaker result than the existence of minimal degrees that can be proved with techniques not much different than the Kleene-Post ones we have seen already. An important generalization (Theorem exactpair 5.2.14) will have many applications. We first introduce a notationally convenient way of dealing with pairs of indices computing from two different sets.

Proof.

posner **Remark 5.2.8 (Posner's trick)** *We are often in a situation where we want to list requirements that involve all pairs of Turing reductions Φ_i and Φ_j with two different sets A and B as oracles. We can save one set of indices by noticing that we can get by in such a listing using just one index e. The point is that we know that A and B are different (say by construction). For definiteness, suppose that $A(x) = 0$ while $B(x) = 1$ for some x. Given any indices i and j, we can find an e such that for any oracle Z, $\Phi_i^Z = \Phi_e^Z$*

if $Z(x) = 0$ and $\Phi_e^Z = \Phi_j^Z$ if $Z(x) = 1$. Using Φ_e for computing from both A and B then gives the same results as using i and j to compute from A and B, respectively. This notational device is known as Posner's trick and we will use it frequently. After the first use in Theorem 5.2.9, we will do so usually, without comment or specific reference to the fact that the sets being constructed are distinct).

■

minpair **Theorem 5.2.9 (Minimal Pair)** *There are $A, B > 0$ such that $\mathbf{a} \wedge \mathbf{b} = \mathbf{0}$, i.e. for all C, if $C \leq_T A, B$ then $C \equiv_T 0$. Note we will often abuse notation and write $A \wedge B \equiv_T 0$ in place of $\mathbf{a} \wedge \mathbf{b} = \mathbf{0}$ and similarly $A \wedge B \equiv_T C$ for $\mathbf{a} \wedge \mathbf{b} = \mathbf{c}$.*

Proof. We build A, B by finite approximations α_s, β_s with union A and B, respectively. There are three kinds of requirements:

$$P_e : \Phi_e \neq A, \qquad Q_e : \Phi_e \neq B \qquad \text{and} \qquad N_e : \Phi_e^A = \Phi_e^B = C \Rightarrow C \text{ is recursive.}$$

Again we order the requirements arbitrarily in a list R_s. Note that we are using Posner's trick to replace the natural requirements $N_{i,j} : \Phi_i^A = \Phi_j^B = C \Rightarrow C$ is recursive by the N_e above.

Construction: We begin with $\alpha_0 = \langle 0 \rangle$ and $\beta_0 = \langle 1 \rangle$ (to make A and B obviously distinct) and, given α_s and β_s we act at stage $s + 1$ to satisfy R_s.

If $R_s = P_e$, ask if $\Phi_e(|\alpha_s|) \uparrow$. If so, the requirement is already satisfied and we do nothing, i.e. we let $\alpha_{s+1} = \alpha_s$ and $\beta_{s+1} = \beta_s$. Otherwise, let α_{s+1} be defined by setting $(|\alpha_s|) = 1 - \Phi_e(|\alpha_s|)$ and let $\beta_{s+1} = \beta_s$. Again the requirement is satisfied. If $R_s = Q_e$, the procedure is the same except that we interchange α and β.

If $R_s = N_e$, ask if $(\exists \alpha \supseteq \alpha_s)(\exists \beta \supseteq \beta_s)(\exists x)(\Phi_e^\alpha(x) \downarrow \neq \Phi_e^\beta(x) \downarrow)$. If such extensions exist, pick the first pair $\langle \alpha, \beta \rangle$ which satisfies the condition and put $\alpha_{s+1} = \alpha$ and $\beta_{s+1} = \beta$. If no such extensions exist, do nothing.

Verification: We have already observed that all the P_e and Q_e are satisfied so $A, B > 0$. For N_e, we may assume that $\Phi_e^A = \Phi_e^B = C$ as otherwise the requirement is automatically satisfied. We want to show that C is recursive. Consider α_s, β_s for the stage s such that $N_e = R_s$. To compute $C(x)$, find any finite extension $\alpha \supseteq \alpha_s$ such that $\Phi_e^\alpha(x)$. (There is one since $A \supseteq \alpha_s$ and $\Phi_e^A(x) \downarrow$.) We claim that $\Phi_e^\alpha(x) = C(x)$. If not, there is a $\beta \supseteq \beta_s$ with $\beta \subseteq B$ such that $\Phi_e^\beta(x) = \Phi_e^B(x) = C(x)$ and so we would have acted at s with α and β (if not another pair of extensions) contrary to our assumption.
■

We frequently use this idea of searching for extensions that give different outputs when used as oracles for a fixed Φ_e and, if we find them, doing some kind of diagonalization. If there are none, we generally argue that Φ_e^A is recursive (or recursive in the relevant notion of extension as in Theorem 5.2.14). We extract the appropriate notion and provide some terminology.

esplit **Definition 5.2.10** *We say that two strings σ and τ* e-split *(or form an* e-splitting*) if $\exists x(\Phi_e^\sigma(x)\downarrow\neq\Phi_e^\tau(x)\downarrow)$. We denote this relation by $\sigma|_e\tau$ and say that σ* and *τ* e-split at x. *Note that by our conventions in ??Definition* useconv *??, $\Phi_e^\sigma(x)=\Phi_{e,|\sigma|}^\sigma(x)$ is a recursive relation as is $\exists x(\Phi_e^\sigma(x)\downarrow\neq\Phi_e^\tau(x)\downarrow)$, i.e. $\sigma|_e\tau$.*

Exercise 5.2.11 *We may make the A and B of Theorem* minpair *5.2.9 low or note that as constructed they are already low. We can also relativize the result: $\forall\mathbf{c}\exists\mathbf{a},\mathbf{b}(\mathbf{a}\wedge\mathbf{b}\equiv C\ \&\ \mathbf{a}'=\mathbf{b}'=\mathbf{c}')$.*

Exercise 5.2.12 *Improve Exercise* coneavoid' *5.2.3 by showing that we can make $B'\leq_T A\oplus 0'$.*

We now want a notion similar to minimal pairs but with an arbitrary countable ideal of degrees playing the role of $\mathbf{0}$.

ideal **Definition 5.2.13** *$\mathcal{C}\subseteq\mathcal{D}$ is an* ideal *in the uppersemilattice $\mathcal{D}$ if it is closed under joins and is closed downwards (i.e. if $\mathbf{y}\in\mathcal{C}$ and $\mathbf{x}\leq\mathbf{y}$ then $\mathbf{x}\in\mathcal{C}$). An ideal $\mathcal{C}$ is* principal *if there is a degree $\mathbf{c}$ such that $\mathcal{C}=\{\mathbf{x}:\mathbf{x}\leq_T\mathbf{c}\}$.*

exactpair **Theorem 5.2.14 (Exact Pair)** *If $\mathcal{C}$ is any countable ideal in $\mathcal{D}$, there are $\mathbf{a},\mathbf{b}$ such that $\mathcal{C}=\{\mathbf{x}:\mathbf{x}\leq_T\mathbf{a},\mathbf{b}\}=\{\mathbf{x}:\mathbf{x}\leq_T\mathbf{a}\}\cap\{\mathbf{x}:\mathbf{x}\leq_T\mathbf{b}\}$.*

This theorem gives a very strong characterization of the countable ideals $\mathcal{C}$ of $\mathcal{D}$. It says that, if not principal, $\mathcal{C}$ is at least the intersection of two principal ideals. An alternative statement of the theorem that will be used in its proof is the following:

exactpair2 **Theorem 5.2.15** *If $C_1\leq_T C_2\leq_T\cdots$ is an ascending sequence, then there are A,B such that $\{X:X\leq_T A,B\}=\{X:\exists n(X\leq_T C_n)\}$.*

Proposition 5.2.16 *Theorem* exactpair2 *5.2.15 implies Theorem* exactpair *5.2.14.*

Proof. We list all the sets D_j with degrees in a given countable ideal $\mathcal{C}$ and then consider the ascending sequence $C_i=\oplus_{j<i}D_j$. If A and B satisfy Theorem exactpair2 5.2.15 for this sequence then we claim that their degrees $\mathbf{a}$ and $\mathbf{b}$ satisfy Theorem exactpair 5.2.14. To see this suppose $\mathbf{d}\in\mathcal{C}$ then some $D_j\in\mathbf{d}$ and so $D_j\leq_T C_{j+1}$ and so by the conditions of Theorem exactpair2 5.2.15, $\mathbf{d}\leq_\mathbf{T}\mathbf{a},\mathbf{b}$ as required. For the other direction, suppose that $X\leq_T A,B$. The conditions of Theorem exactpair2 5.2.15 imply that $X\leq_T C_i=\oplus_{j<i}D_j$ for some i. As $\mathcal{C}$ is closed under finite joins and downwards, $\mathbf{x}\in\mathcal{C}$ as required. ■

Exercise 5.2.17 *Theorem* exactpair *5.2.14 implies Theorem* exactpair2 *5.2.15 so the two formulations are actually equivalent.*

We now prove the second formulation of the theorem and so the first as well.

Proof of Theorem exactpair2 **5.2.15.** Given $\langle C_n\rangle$ ascending in Turing degree, we build A,B such that

- for all n, $C_n \leq_T A, B$ and
- for every $C \leq_T A, B$, $C \leq_T C_n$ for some n.

To prove the theorem is clearly sufficient to constructing sets A and B that satisfy the following requirements:

$$P_n : C_n \leq_T A, B \qquad\qquad N_e : \Phi_e^A = \Phi_e^B = C \Rightarrow \exists n(C \leq_T C_n).$$

As usual we list the requirements as R_s. We build A, B by approximations α_s, β_s. Instead of these being finite strings, however, they are more like matrices. For each approximation γ there are finitely many i (columns of the matrix) such that the value of $\gamma(\langle i, x\rangle)$ is determined for every x. In addition, there are finitely many other numbers $\langle j, y\rangle$ such that γ is defined at $\langle j, y\rangle$.

Construction: As usual we begin with $\alpha_s = \beta_s = \emptyset$, $A = \cup\alpha_s$ and $B = \cup\beta_s$. We describe the construction at stage $s+1$ given α_s and β_s. Suppose $R_s = P_n$. Choose the least i such that both α_s and β_s are undefined at every $\langle i, x\rangle$. Let α_{s+1} (β_{s+1}) be the result of coding C_n into that column of α_s (β_s) and leaving the rest of the approximation unchanged, i.e. $\alpha_{s+1}(\langle i, x\rangle) = C_n(x) = \beta_{s+1}(\langle i, x\rangle)$ for every x and otherwise there are no differences between $\alpha_s(\beta_s)$ and $\alpha_{s+1}(\beta_{s+1})$. This action clearly satisfies P_n

Otherwise, suppose $R_s = N_e$. We ask if $\exists x(\exists\alpha \supseteq \alpha_s)$ $(\exists\beta \supseteq \beta_s)(\Phi_e^\alpha(x) \downarrow = \Phi_e^\beta(x) \downarrow)$ with the domains of α and β being only finitely larger than those of α_s and β_s, respectively. If such extensions exist, let $(\alpha_{s+1}, \beta_{s+1})$ be the least such pair of extensions (in terms of the finite amount of information added to α_s and β_s). If no such extensions exist, do nothing, i.e. $\alpha_{s+1} = \alpha_s$ and $\beta_{s+1} = \beta_s$.

Verifications: We have already noted that A and B satisfy the P_n requirements and so $C_n \leq_T A, B$ for all n. Consider then the requirements $N_{e,i}$. We may assume that $\Phi_e^A = \Phi_e^B = C$ for some C as otherwise the requirement is automatically satisfied. We want to prove that $C \leq_T C_n$ for some n. Consider stage $s+1$ of the construction for s with $R_s = N_{e,i}$ and let n be the largest m such that we have coded C_m into A and B by stage s (by making some column of α_s and β_s equal to C_m as in the construction at stages devoted to P_m). Now, to compute $C(x)$, find any finite extension α of α_s such that $\Phi_e^\alpha(x)\downarrow$. (There is one since $A \supseteq \alpha_s$ and $\Phi_e^A(x)\downarrow$.) We claim that $\Phi_e^\alpha(x) = C(x)$. If not, there is a finite extension β of β_s with $\beta \subseteq B$ such that $\Phi_e^\beta(x) = \Phi_e^B(x) = C(x)$ and so we would have acted at s with α and β contrary to our assumption. The crucial point now is that checking whether $\alpha \supseteq \alpha_s$ is recursive in C_n. As the finitely many columns of α_s which are defined at all elements are each equal to some C_m for $m < n$ and as α_s is only defined at finitely many numbers not in these finitely many columns, this check is clearly recursive in $\oplus_{i<n}C_i$ (using the extra finite information about which column is which C_i and what other information is in α_s) and so in C_n as the sequence C_i is ascending in Turing degree. ■

Now for the first of the important applications of this Theorem.

notlattice **Corollary 5.2.18** *$\mathcal{D}$ is not a lattice.*

Proof. Let C_i be strictly ascending in Turing degree. (Such exist, for example, by Proposition embrecpo 5.1.16 or simply take $C_i = 0^{(i)}$.) Now let A and B be as in Theorem exactpair2 5.2.15. If there were a C whose degree is the meet of those of A and B then $C \leq_T A, B$ and so $C \leq_T C_n$ for some n. In this case, $C <_T C_{n+1} \leq_T A, B$ for a contradiction. ■

Exercise 5.2.19 *A set $\mathcal{S}$ of degrees has a least upper bound if and only if the join of some finite subset of $\mathcal{S}$ is its least upper bound.*

Exercise 5.2.20 *What is a bound on the complexity (degrees) of the A and B of Theorem exactpair2 5.2.15 in terms of the C_n? Does $(\oplus C_n)'$ work? How about a better bound? Consider also the special case that $C_n = 0^{(n)}$.*

<0'notlat **Exercise 5.2.21** *Use the results of the previous exercise and Corollary emb<0' 5.1.23 to show that $\mathcal{D}(\leq \mathbf{0}')$ is not a lattice.*

We now prove a generalization of Theorem coneavoid 5.2.2 that will be the main ingredient from this Chapter in the decision procedure for the 2-quantifier theory of $\mathcal{D}$ in 2qtth 10.4.

extemb **Proposition 5.2.22 (Extensions of Embeddings)** *Given finite partial orders $\mathcal{P} \subset \mathcal{Q}$ with no $x \in Q - P$ below any $y \in P$ such that, for any $q \in Q$ and $p_0, p_1 \leq q \in Q - P$, there is a $p \in P$ with $p_0, p_1 \leq p \leq q$ and an embedding $f : \mathcal{P} \to \mathcal{D}$, there is an extension g of f embedding $\mathcal{Q}$ into $\mathcal{D}$.*

Proof. Let $P = \{p_i | i \leq m\}$, $Q - P = \{q_j | j \leq n\}$, $X_i \in \mathbf{x}_i = f(p_i)$ for $i \leq m$ and $X = \oplus\{X_i | i \leq m\}$. By our assumptions on the $p_i \leq_{\mathcal{Q}} q_j$, there is, for each $j \leq n$, an $i_j \leq m$ such that p_{i_j} is the largest p_i such that $p_i \leq_{\mathcal{Q}} q_j$. We construct Y_j for $j \leq n$ and set $Z_j = \oplus\{Y_k | q_k \leq q_j\} \oplus X_{i_j}$. Our plan is to let $g(q_j) = \deg(Z_j)$. It is clear that $p_i \leq q_j \to X_i \leq_T Z_j$ (as $X_i \leq_T X_{i_j}$) and $q_j \leq q_k \to Z_j \leq_T Z_l$ (as $p_{i_j} \leq p_{i_k}$ and so $X_{i_j} \leq_T X_{i_k}$ while $q_l \leq q_j \to q_l \leq q_k$ and so $\oplus\{Y_l | q_l \leq q_j\} \leq_T \oplus\{Y_l | q_l \leq q_k\}$). (Remember all these sums are finite.)

To guarantee the required Turing inequalities it clearly suffices to satisfy the following requirements for $Y = \oplus\{Y_j | j \leq n\}$, $i, k \leq m$, $j \leq n$ and $e \in \mathbb{N}$:

$$S_{e,i,k} : \text{If } p_i \nleq p_k \text{ then } X_i \neq \Phi_e^{Y \oplus X_j}$$

$$N_{e,j} :\ Y_j \neq \Phi_e^{\oplus\{Y_k | k \neq j \leq n\} \oplus X}$$

We can use the techniques from Theorems coneavoid 5.2.2 to satisfy these requirements using finite binary strings $\theta_{j,s}$ as approximations to $Y_j = \cup\theta_{j,s}$. We list the requirements as R_s.

Construction: As usual we begin with $\theta_{j,s} = \emptyset$ for $j \leq n$. At stage $s+1$ we have $\theta_{j,s}$ and attempt to satisfy requirement R_s. If $R_s = N_{e,j}$ we ask if there are finite extensions θ_k of $\theta_{k,s}$ for $k \neq j$ such that $\Phi_e^{\oplus\{\theta_k | k \neq j \leq n\} \oplus X}(|\gamma_{j,s}|) \downarrow$. If so, we let θ_k form the least such n-tuple and set $\theta_{k,s+1} = \theta_k$ for $k \neq j$ and $\theta_{j,s+1}(|\theta_{j,s}|) = 1 - \Phi_e^{\oplus\{\theta_k | k \neq j \leq n\} \oplus X}(|\gamma_{j,s}|)$. If there are no such θ_k then make no changes. If $R_s = S_{e,i,k}$, we ask if there are extensions θ_j of $\theta_{j,s}$ for $j \le n$ such, that for some x, $X_i(x) \neq \Phi_e^{\oplus\{\theta_j|j\le n\}\oplus X_j}(x)\downarrow$. If so, we choose the least such $(n+1)$-tuple θ_j and set $\theta_{j,s+1} = \theta_j$ for all $j \le n$. If not, we again make no changes.

Verification: The verifications are as before. For $N_{e,j}$ if we act to extend $\theta_{j,s}$ at stage $s+1$ with $R_s = N_{e,j}$, we have clearly diagonalized to satisfy the requirement and if not, then no extension of the θ_k for $k \neq j$ can make $\Phi_e^{\oplus\{\theta_k|k\neq j\le n\}\oplus X}(|\gamma_{j,s}|)\downarrow$. Thus $\Phi_e^{\oplus\{Y_k|k\neq j\le n\}\oplus X}(|\gamma_{j,s}|)\uparrow$ and we also satisfy the requirement. For $R_s = S_{e,i,k}$, suppose that $p_i \not\le p_k$ but $X_i = \Phi_e^{Y\oplus X_j}$ and consider stage $s+1$. Clearly we did not produce a diagonalization at stage $s+1$. We claim that $X_i \le_T X_j$ for a contradiction. To compute $X_i(x)$ look for any extensions θ_j of $\theta_{j,s}$ such that $\Phi_e^{\oplus\{\theta_j|j\le n\}\oplus X_j}(x)\downarrow$. There must be one as $\Phi_e^{Y\oplus X_j}(x)\downarrow$. If $\Phi_e^{\oplus\{\theta_j|j\le n\}\oplus X_j}(x)\downarrow\neq X_i(x)$ we would have taken such a tuple and diagonalized. As the search for such θ_j is recursive in X_j we have computed X_i recursively in X_j as desired. ■

ctblextemb **Exercise 5.2.23** *Extend Proposition 5.2.22 to countable partial orders $\mathcal{P}$ and $\mathcal{Q}$ satisfying the same conditions as in the Proposition. (This is somewhat more intricate than the constructions we have seen so far in that more must be done to code the order on elements of Q into the sets being constructed to extend the given embedding.)*

aleph1emb **Exercise 5.2.24** *Use Exercise 5.2.23 to prove that every partial order of size $\aleph_1$ with the countable predecessor property can be embedded into $\mathcal{D}$. (Of course, this is a construction of length $\aleph_1$. Some facts about extending partial orders to uppersemilattices from Appendix ?? are also useful here.)*

Notes: Theorems 5.2.2 and 5.2.9 and Corollary 5.2.18 are due to Kleene and Post [1954]. Exercises 5.2.5 and 5.2.6 to Shoenfield [1960]. In fact, Exercise 9.2.33 shows that any maximal antichain has size $2^{\aleph_0}$. Sacks [1961] proves Exercise 5.2.7 but Groszek and Slaman [1983] show that it is consistent that $2^{\aleph_0} = \aleph_2$ but there is a maximal independent set of size $\aleph_1$. Theorem 5.2.14 and Exercise 5.2.21 are due to Spector [1956]. Proposition 5.2.22 and Exercises 5.2.23 and 5.2.24 are due to Sacks [1961].??Posner's trick??

5.3 The range of the jump

The Friedberg Jump Inversion (or Completeness) Theorem 5.3.1 says that the only restriction on the jumps of degrees is the obvious one: they must be at least $\mathbf{0}'$. Thus, the theorem characterizes the range of the jump operator as precisely the set of degrees $\ge_T \mathbf{0}'$. The Shoenfield Jump Inversion Theorem (5.3.5) characterizes the range of the jump operator on the degrees below $\mathbf{0}'$. It too consists of everything not ruled out by the trivial restrictions: $\mathbf{a} \le \mathbf{0}' \to \mathbf{a}' \ge \mathbf{0}'$ & $\mathbf{a}'$ is r.e. in $\mathbf{0}'$.

5.3.1 The Friedberg Jump Inversion Theorem

frcomp **Theorem 5.3.1 (Friedberg Jump Inversion Theorem)**

$$\forall C \geq 0' \exists A (A' \equiv_T C \equiv_T A \vee 0').$$

Proof. Let $C \geq_T 0'$. We build A by finite approximations α_s. The properties that we must insure are the following:

- $C \leq_T A'$ (coding C into A')
- $A' \leq_T C$ (keeping A' as low as possible)
- $A' \leq_T A \vee 0'$ (deciding the jump)

Construction: We begin with $\alpha_0 = \emptyset$. At stage $s+1$ we have α_s. We ask if there is an $\alpha \supset \alpha_s$ such that $\Phi^{\alpha}_s(s) \downarrow$. If so, we choose the least such extension α and let $\alpha_{s+1} = \alpha\hat{\ }C(s)$.

Verification: The construction is recursive in C: As $C \geq_T 0'$ it can answer the question we ask at stage s. If the answer is "yes" then it (indeed even the empty set) can find the least witness and C can, of course then compute $C(s)$) to get α_{s+1}. So $\langle \alpha_s \rangle \leq_T C$. Moreover, $A' \leq_T C$ because $s \in A' \Leftrightarrow \Phi^{\alpha_{s+1}}_s(s) \downarrow$ by construction (if $\Phi^A_s(s)$ converges, it is forced to converge by stage $s+1$).

To show that $C \leq_T A'$ and $A' \leq_T A \vee 0'$, it clearly now suffices to prove that $C \leq_T A \vee 0'$. As $C(s) = \alpha_{s+1}(|\alpha_{s+1}|)$, it is enough to show that the construction is recursive in $A \vee 0'$: Given α_s, $0'$ can answer the question asked at stage s and then compute the least witness α. Now we know that $\alpha_{s+1} = \alpha\hat{\ }C(s)$ and so as $A = \cup_s \alpha_s$, $C(s) = \alpha_{s+1}(|\alpha_{s+1}|) = A(|\alpha_{s+1}|)$ and we can compute it and α_{s+1} from A. Thus, $C \leq_T A \vee 0'$. ■

Exercise 5.3.2 *Prove that all pairs of relations between A and B on the one hand and A' and B' on the other not prohibited by the known facts that $A < A'$ and $A \leq_T B \Rightarrow A' \leq_T B'$ are possible. More precisely, consider all the possible pairs of relations ($<_T$, $\leq_T$, $\equiv_T$ and $|_T$) between A and B and between A' and B'. Determine which such pairs of relations can be realized.*

pojumpinv **Exercise 5.3.3 (Jump inversion preserving partial order)** *Prove that given any finite set S of degrees above $0'$ there is a set T of degrees such that $(T, \leq)$ and $(S, \leq)$ are isomorphic as partial orders and the isomorphism is given by the Turing jump operator.*

5.3.2 The Shoenfield Jump Inversion theorem

We now turn our attention to the jump operator acting on the degrees below $\mathbf{0}'$ and prove an analog of the Friedberg Jump Inversion Theorem frcomp 5.3.1. First note that if $A \leq_T 0'$ then $0' \leq_T A' \leq_T 0''$ and A' is r.e. in $0'$ by ??. This imposes an apparent restriction on the jumps of degrees below $\mathbf{0}'$. We say "apparent" because as far as we know now, it might be that all degrees between $\mathbf{0}'$ and $\mathbf{0}''$ are actually r.e. in $\mathbf{0}'$.

Exercise 5.3.4 *Prove that there is an $A \leq 0'$ such that $\forall e\,(A \not\equiv_T W_e)$ and so by relativization a $\mathbf{c}$ between $\mathbf{0}'$ and $\mathbf{0}''$ which is not r.e. in $\mathbf{0}'$.*

Thus the apparent restriction to degrees REA. in $\mathbf{0}'$ is real. It is, however, the only restriction.

Shjumpinv **Theorem 5.3.5 (Shoenfield Jump Inversion Theorem)** *For every $C \geq 0'$ which is r.e. in $0'$, there is an $A \leq_T 0'$ such that $A' \equiv_T C$.*

Proof. We construct A recursively in $0'$ using an enumeration C_s of C which is recursive in $0'$. Our approximations to A are finite binary strings α_s whose union is A. Our requirements are as follows:

$$P_e : \text{If } e \in C \text{ then } A^{[e]} \text{ is cofinite and if } e \notin C \text{ then } A^{[e]} \text{ is finite.}$$

$$N_e : \text{Make } \Phi_e^A(e) \text{ converge if we can.}$$

Construction: We begin with $\alpha_s = \emptyset$. At stage $s+1$ we have α_s. We get $\hat{\alpha}$ extending α_s by making $\hat{\alpha}(\langle e, x\rangle) = 1$ for each $e < s$ with $e \in C_s$ and each x such that $|\alpha_s| \leq \langle e, x\rangle \leq s$ and $\hat{\alpha}(\langle e, x\rangle) = 0$ for all other x with $|\alpha_s| \leq \langle e, x\rangle \leq s$. Next, we ask if there is an $e \leq s$ such that $\Phi_e^{\alpha_s}(e)\uparrow$ and a finite extension α of $\hat{\alpha}$ such that $\Phi_e^{\alpha}(e)\downarrow$ and α satisfies conditions similar to those in our first extension: For $|\hat{\alpha}| \leq \langle e, x\rangle$ and $e \leq s$, $\hat{\alpha}(\langle e, x\rangle) = 1$ if $e \in C_s$ and $\hat{\alpha}(\langle e, x\rangle) = 0$ if $e \notin C_s$. If so, choose the least such α as α_{s+1}. If not, let $\hat{\alpha} = \alpha_{s+1}$.

Verification: First, note that as C_s is uniformly recursive in $0'$, given α_s, $0'$, can certainly compute $\hat{\alpha}$ at stage s. As the question we then ask is Σ_1 (given $\hat{\alpha}$ and C_s), $0'$ can also compute its answer. It can then find α_{s+1} recursively using again only $\hat{\alpha}$ and C_s. Thus the construction is recursive in $0'$ as then is A.

Next, note that (as in ??), for each e, there is a stage s by which $C_s \upharpoonright e+1 = C \upharpoonright e+1$ (and so $C_s \upharpoonright e+1 = C_t \upharpoonright e+1$ for every $t > s$). It is clear from the construction that for $t > \langle e, x\rangle > |\alpha_s|$, $\alpha_t(\langle e, x\rangle) = 1$ if $e \in C$ and $\alpha_t(\langle e, x\rangle) = 0$ if $e \notin C$. Thus we have satisfied the requirement P_e and so, by the Shoenfield Limit Lemma (limitlemma 4.3.9), $C \leq_T A'$.

Finally, we must show that $A' \leq_T C$. Recursively in C we determine if $e \in A'$ and if so a stage s_e by which $\Phi_e^{\alpha_s}(e)\downarrow$. Suppose we have determined $A'(i)$ and the stages s_i for $i < e$. Let s be a stage larger than all the s_i such that $C_s \upharpoonright e+1 = C \upharpoonright e+1$. We can clearly find such a stage recursively in C as C knows the final value of $C \upharpoonright e+1$ and, as it is above $0'$, it can also run the enumeration C_s and the construction until such a stage is found. If $\Phi_e^{\alpha_{s+1}}(e)\downarrow$, then, of course $e \in A'$ and we can take $s_e = s+1$. If not, we claim that $e \notin A'$ and have completed our inductive step as required. If $e \in A'$ then there is an initial segment α of A extending α_s and indeed the $\hat{\alpha}$ defined at stage $s+1$ such that $\Phi_e^{\alpha}(e)\downarrow$. Given our choice of s, our argument above proving that $C \leq_T A'$ shows that if $i \leq e$, $t > s$, and $t > \langle i, x\rangle > |\alpha_s|$, then $\alpha_t(\langle i, x\rangle) = 1$ if $i \in C$ (or equivalently $i \in C_t$) and $\alpha_t(\langle i, x\rangle) = 0$ if $i \notin C$ (or equivalently $i \notin C_t$). Thus α must satisfy the conditions required for us to choose it as α_{s+1} for the desired contradiction. (We cannot act for an $i < e$ at $s+1$ as $s > s_i$ and by assumption $\Phi_e^{\alpha_s}(e)\uparrow$.) ■

trShjumpinv **Exercise 5.3.6** *Strengthen the Shoenfield jump inversion theorem by making $A <_T 0'$.*

We can strengthen the notion of highness as we did that of lowness in Definition 5.1.10:

shigh **Definition 5.3.7** *$A \leq_T 0'$ is* superhigh *if $0'' \leq_{tt} A'$.*

Exercise 5.3.8 *If we take C in the proof of the Shoenfield jump inversion theorem to be $0''$ then the set A constructed in Exercise 5.3.6 is superhigh.*

Notes: Theorem 5.3.1 is due to Friedberg [1957]. It was only a part of his undergraduate thesis; Exercise 5.3.3 to Shore [1988] and Theorem 5.3.5 to Shoenfield [1959]. The main result of Shore [1988] implies that the analog of Exercise 5.3.3 does not hold for the Shoenfield jump inversion theorem. Indeed there are $\mathbf{a}_0$ and $\mathbf{a}_1$ r.e. in and above $\mathbf{0}'$ with join strictly less than $\mathbf{0}''$ such that if $\mathbf{u}_0, \mathbf{u}_1 < \mathbf{0}'$ and $\mathbf{u}_i' = \mathbf{a}_i$ (for $i = 0, 1$) then $\mathbf{u}_0 \vee \mathbf{u}_1 = \mathbf{0}'$.

Treec 5.4 Trees and sets of size the continuum

So far we have directly constructed only finite or countable classes of sets or degrees. (Some of the exercises use Zorn's Lemma or iterations of length $\aleph_1$ of basically countable constructions to construct sets of degrees of size $\aleph_1$.) We now want to provide some direct constructions of uncountable sets of degrees, indeed ones of size the continuum, with various properties such as being an antichain. Our basic approach is to construct a perfect binary tree T such that $[T]$, the paths of T, constitute the desired set. As in Definition 4.2.1, binary trees T are subsets of $2^{\mathbb{N}}$ and so $[T]$ is a class of sets. As in Definition ??, we say that a binary tree T is perfect if every node has two incomparable successors. The crucial fact is Proposition ?? that there are continuum ($2^{\aleph_0}$ and so uncountably) many paths through every perfect binary tree.

cmanyincom **Theorem 5.4.1** *There is a set of pairwise incomparable degrees of size continuum, $2^{\aleph_0}$.*

Proof. We build a perfect binary tree T such that if $A, B \in [T]$, $A \neq B$, then $A|_T B$.

The requirements on the tree are

$$R_e : \qquad (\forall A \neq B \in [T])(\forall e)(\Phi_e^A \neq B).$$

To meet these requirements, we construct T by finite approximations T_s which are finite binary trees such that every nonmaximal node has two incomparable extensions in T_s. At the end $T = \cup T_s$.

Construction: We begin with $T_0 = \{\emptyset\}$. At the beginning of stage $s+1$, we have a finite binary tree T_s with n many maximal nodes such that every nonmaximal node has two incomparable extensions in T_s. Let $\sigma_0, \ldots \sigma_{n-1}$ be the maximal nodes in T_s. Any path through the final tree T has one of these as an initial segment. We want to define T_{s+1} so as to satisfy R_s for sets A and B extending incomparable maximal nodes in T_s and to give each σ_i two incomparable successors in T_{s+1}. We list the pairs $\langle i, j \rangle$ with $i, j < n$ and $i \neq j$ as $\langle i_k, j_k \rangle$ for $0 < k \leq l$ and define, by induction on k, $\sigma_{i,k}$ for $i < n$, $0 < k \leq l$ so as to satisfy R_s for sets A and B extending $\sigma_{i_k,k}$ and $\sigma_{j_k,k}$ respectively. We begin with all $\sigma_{i,0} = \sigma_i$ and then assume we have defined the $\sigma_{i,k}$ (by induction). We ask if there is a $\hat{\sigma} \supseteq \sigma_{i_k,k}$ such that $\Phi_s^{\hat{\sigma}}(|\sigma_{j_k,k}|) \downarrow$. If so, we let $\sigma_{i_k,,k+1}$ be the least such $\hat{\sigma}$ and let $\sigma_{j_k,k+1}$ extend $\sigma_{j_k,k}$ by setting $\sigma_{j_k,k+1}(|\sigma_{j_k,k}|) = 1 - \Phi_s^{\sigma_{i_k,,k+1}}(|\sigma_{j_k,k}|)$. We let $\sigma_{i,k+1} = \sigma_{i,k}$ for $i \neq i_k$ or j_k. If there is no such $\hat{\sigma}$, we let $\sigma_{i,k+1} = \sigma_{i,k}$ for all i. Finally we add $\sigma_{i,l}\hat{\ }0$ and $\sigma_{i,l}\hat{\ }1$ (and all their initial segments) to T_s to get T_{s+1}.

Verifications: It is clear from the construction that $T = \cup T_s$ is a perfect binary tree. Moreover, it is clear that if $A \in [T]$ extends some maximal node σ_i of T_s then (with the notation as in the construction) it extends all the $\sigma_{i,k}$ for $k \leq l$ constructed at stage $s+1$. Suppose A and B are distinct paths in T and consider requirement R_e. As $A \neq B$ there is clearly a t such that for every $s \geq t$ the maximal nodes on T_s that are initial segments of A and B are distinct. By the Padding Lemma padding 2.1.12 there is an $s > t$ such that $\Phi_s = \Phi_e$. We claim that at stage $s+1$ of the construction we guaranteed that $\Phi_e^A \neq B$ as required. To see this, let σ_i and σ_j be the maximal nodes of T_s which are initial segments of A and B, respectively and let k be such that at stage $s+1$ we have $\langle i_k, j_j \rangle = \langle i, j \rangle$. As we noted A and B extend $\sigma_{i,k+1}$ and $\sigma_{j,k+1}$ respectively. If at substage $k+1$ of stage $s+1$ we properly extended $\sigma_{j,k}$ to get $\sigma_{j,k+1}$ then $\Phi_s^{\sigma_{i,k+1}}(|\sigma_{j_k,k}|) \downarrow \neq \sigma_{j,k+1}(|\sigma_{j_k,k}|)$ and so $\Phi_e^A(|\sigma_{j_k,k}|) \downarrow = \Phi_s^A(|\sigma_{j_k,k}|) \downarrow \neq B(|\sigma_{j_k,k}|)$ as required. If we did not so extend $\sigma_{j,k}$ then there is no extension $\hat{\sigma}$ of $\sigma_{i,k}$ such that $\Phi_s^{\hat{\sigma}}(|\sigma_{j_k,k}|) \downarrow$ and so none such that $\Phi_e^{\hat{\sigma}}(|\sigma_{j_k,k}|) \downarrow$. In particular, $\Phi_e^A(|\sigma_{j_k,k}|) \uparrow$ and we again have satisfied the requirement. ■

Exercise 5.4.2 *There is a size continuum set of degrees which are pairwise minimal, more precisely, there is a perfect binary tree T such that, for $A \neq B \in [T]$, $A, B >_T 0$ and $\forall C(C \leq_T A, B \rightarrow C \equiv_T 0)$.*

Exercise 5.4.3 *There is an independent set of degrees of size continuum.*

These results give many more embedding theorems for uncountable partial orders.. Exercise alephiemb 5.2.24 shows that any size $\aleph_1$ partial order with the countable predecessor property can be embedded into $\mathcal{D}$. It is still an open question if every size continuum partial order with the countable predecessor property can be embedded into $\mathcal{D}$.

Notes: The results presented in this section follow from ones in Sacks [1961].

Chapter 6

Forcing in Arithmetic and Recursion Theory

forcing

6.1 Notions of Forcing and Genericity

Forcing provides many generalizations of the techniques we have developed in Chapter embeddings 5 along with a common language for them all. It captures the idea of approximation to a desired object and how individual approximations guarantee (force) that the object we are building satisfies some requirement. Now approximations usually come with some sense of when one is better, or gives more information, than another. Of course, an approximation may have improvements which are incompatible with each other, i.e. the set of approximations is partially ordered. The intuition is that $p \leq q$ means that p refines, extends or has more information than q. We are generally thinking that the conditions are approximations to some object $G : \mathbb{N} \to \mathbb{N}$ (typically a set) and that if $p \leq q$ then the approximation p gives more information than q and so the class of potential objects that have p as an approximation is smaller then the one for q. In addition, we have some notion of what, at least at a basic level, the approximation p says about G. We formalize these ideas in the rest of this section.

forcing1 **Definition 6.1.1** *A* notion of forcing *is a partial order $\mathcal{P}$ with domain a set P and binary relation $\leq_{\mathcal{P}}$. We call an element of $\mathcal{P}$ a* (forcing) condition*. For convenience, we assume that the partial order has a greatest element* **1***. (For further restrictions see Definition forcing2 6.1.12.) We often write $\leq$ for $\leq_{\mathcal{P}}$ and confuse the underlying set P with the partial order $\mathcal{P}$ when the notion of forcing being used is clear from the context.*

cohenfor **Example 6.1.2** *If the notion of forcing is $(2^{<\omega}, \supseteq)$, then $\sigma \leq \tau \equiv \sigma \supseteq \tau$. In many of our previous constructions we used such binary strings σ as approximations to a set G such that $\sigma \subset G$. So the longer the string, the fewer sets that "satisfy" it, i.e. have it as an approximation (initial segment). This example is often called* Cohen forcing*. Common, but essentially equivalent (See Exercise cohensame* **??***), variations include $(\mathbb{N}^{<\omega}, \supseteq)$ ($(k^{<\omega}, \supseteq)$ for $k \in \mathbb{N}$) approximating a function from $\mathbb{N}$ to $\mathbb{N}$ ($\mathbb{N}$ to $\{0, 1, 2\}$)and the set of finite partial* *maps from $\mathbb{N}$ into $\mathbb{N}$ or from $\mathbb{N}$ into 2 (or k) ordered by extension approximating functions ($\mathbb{N} \to \mathbb{N}$) or sets (or from $\mathbb{N}$ into k), respectively . These types of forcing notions with finite conditions play a central role in this Chapter and the next.*

Example 6.1.3 *In §Treec 5.4 we used finite binary trees with extension while requiring that the tree extension add only strings that are extensions (as strings) of leaves of the given tree. The object being approximated was a binary tree T.*

xactpairfor **Example 6.1.4** *In Theorem exactpair2 5.2.15, we used partial characteristic functions α defined on some finite set of columns and some finitely many additional points. Again we were approximating a set $G \supset \alpha$.*

spectorfor **Example 6.1.5** *Another important notion of forcing is the set of perfect recursive binary trees with $S \leq T \Leftrightarrow S \subseteq T$. (Here trees T are simply sets of binary strings as in Definition tree 4.2.1. The tree is perfect if every node has incomparable extensions as in Definition perfect ????.) We are thinking of such a tree T as approximating some path on T. So refinement (more information) means fewer possible paths, i.e. more information about which path is being approximated. This notion of forcing is often called* Spector forcing *(or* perfect forcing *or* Sacks forcing*). Again there are many variants. For example the trees can be n-branching for some fixed n or finitely branching ?? with the number of branches ?? specified recursively. These notions of forcing play a central role in our constructions of initial segments of $\mathcal{D}$ in Chapters mindeg 9 and initialseg 10.*

How, in general, can we specify what object is it exactly or what class of potential objects is it that a condition p in a notion of forcing approximates? For Cohen forcing a condition (string) σ approximates the class of sets $\{G | G \supset \sigma\}$. So the collection of all approximations to a single set G is simply $\{\sigma | \sigma \subset G\}$, the class of all the initial segments of G. We want to isolate the salient features of this set of conditions or any set $\mathcal{G} \subseteq \mathcal{P}$ that might be considered as an object its members are approximating. The general approach that we want for an arbitrary notion of forcing begins with filters.

Perhaps the most salient feature of a subset $\mathcal{F}$ of $\mathcal{P}$ being an approximation to something is that it not contain contradictory information so that there is, in the end, some object for which we can view every member of the set as an approximation. We do this by requiring that for every two elements of $\mathcal{F}$ there is a third that extends each of the two given ones.

compat **Definition 6.1.6** *Two conditions p, q of a notion of forcing $\mathcal{P}$, are* compatible *if and only if $\exists r(r \leq p \wedge r \leq q)$. If p, q are incompatible we write $p \perp q$ (as opposed to $p \mid q$ which we use to denote* incomparability, *$p \nleq q$ and $q \nleq p$).*

filter **Definition 6.1.7** *$\mathcal{F} \subseteq \mathcal{P}$ is a filter on $\mathcal{P}$ if and only if $\mathcal{F}$ is nonempty, upward closed and, for every $p, q \in \mathcal{F}$, there is an $r \in \mathcal{F}$ with $r \leq p, q$ (so p and q are compatible with a witness in $\mathcal{F}$).*

We are thinking of filters as connected by some procedure to the object we are approximating.

Example 6.1.8 *Suppose we want to approximate a set $G \in 2^{\mathbb{N}}$ and our notion of forcing is $(2^{<\omega}, \supseteq)$ (finite binary strings). Then the set $\{\sigma : \sigma \subset G\}$ is a filter. In particular, the union of this set (filter) is the characteristic function G. It will commonly be the case that the object we want is defined from a filter by some "simple" operation such as union. We formalize this idea in Definition forcing2 6.1.12. Note that for finite strings, being comparable is the same as being compatible.*

Example 6.1.9 *Suppose we want to approximate a set $G \in 2^{\mathbb{N}}$ and our notion of forcing $\mathcal{P}$ is some countable set of infinite binary trees (not necessarily perfect) such as the recursive ones. Then the subset $\{T : G \in [T]\} = \{T : \forall \sigma \subset G(\sigma \in T)\}$ of P is a filter: Suppose two trees both have G as a path. Then the tree which is the (set) intersection of the two trees is a common refinement. For upward closure, if G is a path on T and $T \subseteq S$ then G is also a path on S. In this case, the intersection of (the trees in) this filter is the characteristic function of G.*

For a given notion of forcing $\mathcal{P}$, there is often a canonical way to associate some set or function with a filter $\mathcal{F}$ on $\mathcal{P}$. For example, for Cohen forcing we can naturally try $\cup\mathcal{F}$. For forcing with binary trees we might try $\cap\{[T]|T \in \mathcal{F}\}$. Does this always make sense even for Cohen or Spector forcing? For Cohen forcing it might be that $\cup\mathcal{F}$ is a finite string so itself a condition. For Spector forcing $\cap\{[T]|T \in \mathcal{F}\}$ could be a set of paths through a binary tree with more than one branch which might not necessarily be recursive or perfect. We need to add conditions on our filter to make sure we get a total function or a single set at the end. We might for example require for Cohen forcing that $\mathcal{F}$ contain strings of every (equivalently arbitrarily long) length, i.e. $(\forall n)(\exists \sigma \in \mathcal{F})(|\sigma| \geq n)$. For Spector forcing we could require that there are trees in $\mathcal{F}$ with arbitrarily long nodes σ before the first branching (i.e. σ has two immediate successors in the tree but no $\tau \subset \sigma$ does).

In general, we want to assure that we can associate a unique object to the set of approximations (filters) that we consider. We then want these objects to witness various theorems asserting the existence of objects with prescribed properties. So we want a notion of forcing with conditions approximating the objects asserted to exist in the theorem. As in Chapter embeddings 5, we then usually have requirements that, if met, make the objects constructed satisfy the given theorem. Our desire for the set of approximations to specify a unique object can then be seen as simply another (albeit very basic) goal of the construction to be satisfied by meeting various requirements. (See, for example, Question nonterm 5.1.2.) Our constructions in Chapter embeddings 5 typically worked because at any stage with a given approximation to our desired set we could extend the approximation so as to satisfy any requirement. This property of approximations and requirements is a type of density in the partial order of approximations. Thus, for each of these requirements, we want the set of conditions guaranteeing (forcing) that we satisfy the requirement to be dense. Meeting them all produces a "generic" object which provides a witness for the given theorem. The requirements for the uniqueness of the object approximated at the end is handled in the same way. We now formalize some of these ideas.

dense **Definition 6.1.10** *A subset D of $\mathcal{P}$ is* dense in $\mathcal{P}$ *if* $\forall p \in \mathcal{P} \exists q \in \mathcal{D}(q \leq_P p)$. *A subset D of $\mathcal{P}$ is* dense below $r \in \mathcal{P}$ *if* $\forall p \leq r \exists q \in \mathcal{D}(q \leq p)$.

Cgen **Definition 6.1.11** *If $\mathcal{C}$ is a class of dense subsets of $\mathcal{P}$, we say that $\mathcal{G} \subseteq \mathcal{P}$ is* C-generic *if $\mathcal{G} \cap D \neq \emptyset$ for all $D \in \mathcal{C}$. We say that a sequence $\langle p_n \rangle$ of conditions is $\mathcal{C}$-generic if $\forall i(p_{i+1} \leq_{\mathcal{P}} p_i)$ and $\forall D \in \mathcal{C} \exists n(p_n \in D)$.*

For the specific goal of constructing a uniquely specified object, we add to every notion of forcing a function $V(p)$ representing the atomic information about our generic object determined by the condition p and the requirement that all generic filters meet certain dense sets defined in terms of V.

forcing2 **Definition 6.1.12** *From now on, we require that every notion of forcing have a* valuation function $V : P \to \mathbb{N}^{<\omega}$ *which is recursive on $\mathcal{P}$ and order preserving in the sense that if $p \leq_{\mathcal{P}} q$ then $V(p) \supseteq V(q)$. (We say that a partial recursive function Φ is* recursive on a set X *if* $X \subseteq \mathrm{dom}(\Phi)$.*) Moreover, we require that the sets $V_n = \{p| \, |V(p)| \geq n)\}$ are dense. We also require that any collection of dense sets that we consider for the construction of a generic filter or sequence includes the V_n.*

Vcomp **Exercise 6.1.13** *If, for two forcing conditions p and q, $V(p)$ and $V(q)$ are incompatible as elements of $\mathbb{N}^{<\omega}$, then $p \perp q$. The converse does not hold for every notion of forcing.*

Proposition 6.1.14 *With the conventions of Definition* forcing2 *6.1.12 now in place, for every $\mathcal{C}$-generic sequence $\langle p_n \rangle$ (or filter $\mathcal{G}$), $\cup V(p_n)$ (or $\{V(p)|p \in \mathcal{G}\}$, is a total function.*

Proof. The definitions of a valuation function V as order preserving and of generic sequences (or filters), guarantee that the $V(p_n)$ (or $V(p) \in \mathcal{G}$) are pairwise comparable strings. (See Exercise Vcomp 6.1.13.) The requirement that the V_n are each dense and met by $\langle p_n \rangle$ (or $\mathcal{G}$) make $\cup V(p_n)$ (or $\{V(p)|p \in \mathcal{G}\}$ total. ■

We now see how to specify the unique function associated with each generic sequence or filter.

assocgendef **Definition 6.1.15** *We* associate *to each $\mathcal{C}$-generic sequence $\langle p_n \rangle$ or filter $\mathcal{G}$ the* generic function $G = \cup V(p_n)$ *or* $\{V(p)|p \in \mathcal{F}\}$.

Example 6.1.16 *In Example* cohenfor *6.1.2 for basic Cohen forcing, we may define $V(p) = p$. In Example* spectorfor *6.1.5 (Spector forcing), we may let $V(p)$ be the largest σ such that every $\tau \in T$ is comparable with σ.*

Exercise 6.1.17 *Show that the corresponding V_n are dense for Cohen and Spector forcing.*

Exercise 6.1.18 *What could V be for the variations of Cohen forcing of Example* cohenfor *6.1.2? Prove that the corresponding V_n are dense.*

Exercise 6.1.19 *What could V be for the forcing of Example* exactpairfor *6.1.4 that constructs an exact pair? Prove that the corresponding V_n are dense.*

seqfilter **Proposition 6.1.20** *If $\langle p_n \rangle$ is a $\mathcal{C}$-generic sequence then $\mathcal{G} = \{p | \exists n (p_n \leq p\}$ is a $\mathcal{C}$-generic filter containing each p_n.*

Proof. $\mathcal{G}$ is $\mathcal{C}$-generic because it contains an element, p_n, of D_n for all n. It is upward closed because if $p \in \mathcal{G}$ then $p \geq p_e$ for some e so if $q > p \geq p_e$ then $q \geq p_e$ as well. Finally, it is pairwise compatible because given $p \geq p_{e_1}$, $q \geq p_{e_2}$ then $p, q \geq p_e$ where $e = \max\{e_1, e_2\}$. ■

If our collection of dense sets is countable (as it always essentially is in our applications) then generic sequences, filters and functions always exist.

ECgen **Theorem 6.1.21** *If $\mathcal{C}$ is countable and $p \in \mathcal{P}$, then there is a $\mathcal{C}$-generic sequence $\langle p_n \rangle$ with $p_0 = p$ and so, by Proposition* seqfilter *6.1.20, a $\mathcal{C}$-generic filter $\mathcal{G}$ containing p.*

Proof. Let $\mathcal{C} = \{D_n | n \in \mathbb{N}\}$. We define $\langle p_n \rangle$ by recursion beginning with $p_0 = p$. If we have p_n then we choose any $q \leq p_n$ in D_n as p_{n+1}. One exists by the density of D_n. It is clear that $\langle p_n \rangle$ is a $\mathcal{C}$ generic sequence and so $\mathcal{G} = \{p | \exists n (p_n \leq p\}$ is $\mathcal{C}$-generic filter containing p. ■

filterseq **Exercise 6.1.22** *If $\mathcal{C}$ is countable, each $D_n \in \mathcal{C}$ is downward closed (as we can almost always guarantee) and $\mathcal{G}$ is a $\mathcal{C}$-generic filter containing p, then there is a $\mathcal{C}$-generic sequence $\langle p_n \rangle$ with $p_0 = p$ such that $\mathcal{G} = \{p | \exists n (p_n \leq p\}$. (This is a partial converse to Proposition* seqfilter *6.1.20.)*

As is our general practice, we often want to know how hard it is to compute a $\mathcal{C}$-generic sequence, filter or function. We must begin with the complexity of $\mathcal{P}$ and then consider how hard it is to compute the generic sequence $\langle p_n \rangle$ and so the associated generic G. We view the elements of $\mathcal{P}$ as being (coded by) natural numbers. For convenience we let the natural number 1 be the greatest element of P. Finding the generic G from the sequence, requires only knowing the set P.

assocgen **Proposition 6.1.23** *If G is associated with the $\mathcal{C}$-generic sequence $\langle p_n \rangle$ (filter $\mathcal{G}$) then $G \leq_T \langle p_n \rangle \oplus P$ ($\mathcal{G} \oplus P$).*

Proof. As V is recursive on P and the sets V_n of Definition forcing2 6.1.12 are included in $\mathcal{C}$, we can, recursively in P, compute $G(n)$ by searching for a $p \in V_n$ such that $p = p_k$ for some k (or $p \in \mathcal{G}$) and then noting that $G(n) = V(p)$. ■

To compute a generic sequence, we usually need the order relation $\leq_{\mathcal{P}}$ as well as a procedure for finding extensions in the given dense sets.

densityf **Definition 6.1.24** *A notion of forcing $\mathcal{P}$ is* A-recursive *(or* **a**-recursive*) if the set P and the relation $\leq_{\mathcal{P}}$ are recursive in A ($\in$ **a**). (As usual if $A = \emptyset$ (**a** $=$ **0**) we omit it from the notation.) If $\mathcal{C} = \{D_n\}$ is a collection of dense sets in $\mathcal{P}$ then f is a* density function *for $\mathcal{C}$ if $(\forall p \in P)(\forall n \in \mathbb{N})(f(p,n) \in D_n)$.*

meetdense **Proposition 6.1.25** *If $\mathcal{P}$ is an A-recursive notion of forcing and $\mathcal{C} = \{D_n\}$ is a uniformly A-recursive sequence of dense subsets of $\mathcal{P}$ and $p \in P$ then there is a $\mathcal{C}$-generic sequence $\langle p_n \rangle$ with $p_0 = p$ which is recursive in A. More generally, for an arbitrary notion of forcing $\mathcal{P}$, $p \in P$ and a class $\mathcal{C} = \{D_n\}$ of dense sets, if f is a density function for $\mathcal{C}$, then there is a $\mathcal{C}$-generic sequence $\langle p_n \rangle \leq_T f$ with $p_0 = p$. The generic G function associated with these filters or sequences are also recursive in A or $f \oplus A$, respectively.*

Proof. If $\mathcal{P}$ is an A-recursive notion of forcing and $\mathcal{C} = \{D_n\}$ is a uniformly A-recursive sequence of dense subsets of $\mathcal{P}$, then we can define a density function $f \leq_T A$ by letting $f(p,n)$ be the least (in the natural order of $\mathbb{N}$) $q \leq_{\mathcal{P}} p$ with $q \in D_n$. In either case, the desired generic sequence is now given by setting $p_0 = p$ and $p_{n+1} = f(n, p_n)$. That G is recursive in A or $f \oplus A$ now follows from Proposition assocgen 6.1.23. ■

Note that the generic filter $\mathcal{G}$ defined from the generic sequence $\langle p_n \rangle$ in Proposition seqfilter 6.1.20 is Σ_1 in $\langle p_n \rangle$ but not necessarily recursive in it (Exercise filternotrec 6.1.26). In the other direction, there is usually some generic sequence recursive in the filter (Exercise filterrec 6.1.28).

ilternotrec **Exercise 6.1.26** *Give a recursive notion of forcing and a $\{V_n\}$-generic sequence $\langle p_i \rangle$ such that $\mathcal{G} = \{p | \exists i (p_i \leq p)\}$ is not recursive.*

??Hint:$P = \{p_{i,j} | i, j \in \mathbb{N}\}$. $p_{i,j} \leq_{\mathcal{P}} p_{i,k}$ for $j \geq k$ and all i and for $i \neq 0$, $p_{0,n} \leq_{\mathcal{P}} p_{i,0}$ if $i \in K_n$ and not otherwise. $V(p_{i,j}) = 1^j$. $\langle p_i \rangle = \langle p_{0,i} \rangle$. ??

Exercise 6.1.27 *Show that for a recursive notion of forcing $\mathcal{P}$ for which the relation $\perp$ is also recursive and any collection of dense sets $\mathcal{C}$ that include $D_q = \{p | p \leq q$ or $(\forall r \leq p)(r \not\leq q)\}$ for every $q \in P$ and every $\mathcal{C}$-generic sequence $\langle p_n \rangle$, the generic filter $\mathcal{G} = \ \{p | \exists n (p_n \leq p)\}$ is recursive in $\langle p_n \rangle$.*

filterrec **Exercise 6.1.28** *Show that in Exercise filterseq 6.1.22, if the notion of forcing is A-recursive then we may take the sequence $\langle p_n \rangle$ to be recursive in $\mathcal{G} \oplus A$.*

[There are various connections between forcing, (generic) filters and topology. Order topology on $\mathcal{P}$...dense open sets , meager comeager, generic..

In Cohen forcing the conditions correspond to (approximate) open sets in Cantor space $2^{\mathbb{N}}$ i.e. σ is an approximation to each set $G \supset \sigma$ and these form an open (even clopen) set in $2^{\mathbb{N}}$. Then the intersection of all the clopen sets in a filter $\mathcal{F}$ is a closed set. If the filter is mildly generic it is the single set G which is the union of the filter. In Spector forcing the intersection of the $[T]$ for T in some filter is a closed set. It is nonempty since the space is compact. If the filter is mildly generic the intersection is also a singleton.]

.

6.2 The Forcing Language and Deciding Classes of Sentences

The ad hoc approach to constructions presented in Chapter embeddings 5 looks at a theorem we want to prove, decides what are the specific requirements we need to meet, what approximations we should use and how to extend approximations to satisfy the requirements. It then builds the desired sets accordingly. For example, this is what we did to build two Turing incomparable sets $A|_T B$. The requirements were $\Phi_e^A \neq B$ (and $\Phi_e^B \neq A$). Our approximations were pairs $\langle\alpha,\beta\rangle$ of binary strings. Given $\langle\alpha,\beta\rangle$, we could find $\langle\hat{\alpha},\hat{\beta}\rangle \leq \langle\alpha,\beta\rangle$ which would guarantee any particular requirement. In particular, if there is an x and a pair of extensions $\langle\hat{\alpha},\hat{\beta}\rangle$ of $\langle\alpha,\beta\rangle$ such that $\exists x(\Phi_e^{\hat{\alpha}}(x)\downarrow \neq \hat{\beta}(x)\downarrow)$, we chose the least one; if not, we took $\langle\alpha,\beta\rangle$ itself as the next approximation. In the terminology of forcing, we have a notion of forcing with conditions pairs $\langle\alpha,\beta\rangle$ and refinement given by extension. Meeting the requirements $\Phi_e^A \neq B$ corresponds to getting into the dense sets

$$D_e = \{\langle\alpha,\beta\rangle : \exists x(\Phi_e^{\alpha}(x)\downarrow \neq \beta(x)\downarrow) \text{ or } (\forall\langle\hat{\alpha},\hat{\beta}\rangle \leq \langle\alpha,\beta\rangle)(\neg[\exists x\Phi_e^{\hat{\alpha}}(x)\downarrow \neq \hat{\beta}(x)\downarrow])\}.$$

Likewise, we defined dense sets C_e, which guaranteed that $\Phi_e^B \neq A$. Now if $\langle\alpha_n,\beta_n\rangle$ is a $\{D_e, C_e\}$-generic sequence and $G_0 = \cup\alpha_n$, $G_1 = \cup\beta_n$, then $G_0 \mid_T G_1$.

In this manner, each of the proofs we did earlier by constructions with requirements can be translated into a notion of forcing (consisting of the approximations with a natural order), dense sets and generics for the dense sets D_e determined by the conditions that guarantee (force) that we satisfy the eth requirement. However, a primary benefit of the forcing technology is the generality it allows. For example, we can tackle many of the constructions simultaneously and so give one theorem implying many of our previous ones that were proven individually.

To this end, we need to define the forcing relation ($\Vdash$) more generally, by induction on formulas φ that somehow say that if $p \Vdash \varphi$ then $\varphi(G)$ holds for the set or function G determined by any sufficiently generic sequence $\langle p_n\rangle$ or filter $\mathcal{G}$. Thus we want a relation $\Vdash$ between conditions $p \in \mathcal{P}$ and sentences $\phi(\mathtt{G})$ (where we use $\mathtt{G}$ as the formal symbol that is to be interpreted as our generic set or function G). This relation should approximate truth in the sense just described. We could use a standard language of arithmetic as in arithh §4.4 (in set theoretic forcing, one would use the language of set theory) augmented with another parameter ($\mathtt{G}$) for the set we are building. (At times we may also want other fixed other parameters ($\bar{h}$) for given functions. As usual we typically leave this case to relativization.)

The usual definition of forcing adopts some such standard language (in our case, for arithmetic) and proceeds by induction on the full range of formulas with the crucial steps (after the atomic variable free formulas) being $p \Vdash \exists x\psi \Leftrightarrow \exists n\,(p \Vdash \varphi(n))$; $p \Vdash \neg\varphi \Leftrightarrow \forall q \leq p(q \nVdash \varphi)$ and so $p \nVdash \forall x\varphi \Leftrightarrow \forall n\forall q \leq p\,(q \nVdash \neg\varphi(n)) \Leftrightarrow \forall n\forall q \leq p\exists r \leq q\,(r \Vdash \varphi(n))$. (The definitions for conjunction and disjunction are given by $p \Vdash \varphi \wedge \psi \Leftrightarrow p \Vdash \varphi$ and $p \Vdash \psi$ and $p \Vdash \varphi \vee \psi \Leftrightarrow p \Vdash \varphi$ or $p \Vdash \psi$.)

For our purposes it is more convenient to use the master (universal) partial recursive functions described and defined in §codeTM 2.1. In particular we want to use the predicate $\varphi_{1+n}(f, e, \bar{x}, s)\downarrow$ that means that Turing machine e, with oracle f and input $\bar{x}$ of length n, converges when run for s many steps. We also exploit the normal form theorems for sentences of arithmetic of §arithh 4.4. With our restricted language we can simplify the definition of forcing and so, crucially, the calculation of the complexity of the relation $p \Vdash \varphi$.

We begin with an arbitrary Σ_1 formula of arithmetic with a set parameter G: $\psi(G, \bar{c}, \bar{x})$. For later convenience we have displayed the numerical terms $\bar{c}$ and free variables $\bar{x}$ appearing in ψ (with $|\bar{c}| = k$ and $|\bar{x}| = l$). By the m-completeness of G' for the class of Σ_1^G sets (Theorem hierarchy 4.5.1), ψ is equivalent to $\exists s(\varphi_{1+n}(G, e, e, \bar{c}, \bar{x}, s)\downarrow)$ (in the usual sense that for every instantiation $\bar{n}$ of the $\bar{x}$, $\psi(G, \bar{c}, \bar{n}) \Leftrightarrow \exists s(\varphi_{1+k+l}(G, e, e, \bar{c}, \bar{n}, s)\downarrow)$). Here e is given as a recursive function of (the code for) the formula ψ, say $e(\psi)$. (The uniformity of the completeness result (??sigmalcomp ????) shows that $e(\psi)$ is independent of the choice of G and that it depends uniformly on $\bar{c}$.) Note that by the analysis and conventions of §codeTM 2.1, this last formula (and so $\psi(G, \bar{c}, \bar{x})$) is equivalent to $(\exists\sigma \subset G)(\varphi_{1+k+l}(\sigma, e, e, \bar{c}, \bar{x})\downarrow)$. Of course, any Π_1 sentence $\hat{\psi}$ is then equivalent to ones of the form $\forall s\neg(\varphi_{1+k+l}(G, e, e, \bar{c}, \bar{x}, s)\downarrow)$ and $(\forall\sigma \subset G)(\neg(\varphi_{1+k+l}(\sigma, e, e, \bar{c}, \bar{x})\downarrow))$.

The normal form theorems of §arithh 4.4, say that any sentence of arithmetic with set parameter G is equivalent to one of the form $\bar{Q}\bar{x}\psi(G, \bar{c}, \bar{x},)$ where $\bar{Q}\bar{x}$ is a string of alternating quantifiers ending with $\forall$ and $\psi(G, \bar{c}, \bar{x})$ is a Σ_1 formula or $\bar{Q}\bar{x}$ is a string of alternating quantifiers ending with $\exists$ and $\psi(G, \bar{c}, \bar{x})$ is a Π_1 formula. So every sentence of arithmetic with one function parameter G is equivalent to a sentence of the form $\bar{Q}\bar{x}\exists s(\varphi_{1+k+l}(G, e, e, \bar{c}, \bar{x}, s)\downarrow)$ or $\bar{Q}\bar{x}\forall s(\neg(\varphi_{1+k+l}(G, e, e, \bar{c}, \bar{x}, s)\downarrow))$. Thus we can inductively define the forcing relation only for such sentences and still be able to talk about all arithmetic sentences.

We need some notational conventions.

Notation 6.2.1 *By the negation $\neg\varphi$ of a formula in this form we mean the natural equivalent gotten by driving the negation sign through the quantifiers, i.e. change each quantifier in $\bar{Q}\bar{x}$ ($\exists$ to $\forall$ and vice versa) and $\exists s(\varphi_{1+k+l}(G, e, e, \bar{c}, \bar{x}, s)\downarrow)$ to its negation $\forall s(\neg(\varphi_{1+k+l}(G, e, e, \bar{c}, \bar{x}, s)\downarrow))$ (and vice versa). For example,*

$$\neg\exists x\forall y\exists s(\varphi_{1+|\bar{c}|+2}(G, e, e, \bar{c}, x, y, s)\downarrow) = \forall x\exists y\forall s(\neg(\varphi_{1+|\bar{c}|+2}(G, e, e, \bar{c}, x, y, s)\downarrow)).$$

So, in particular, $\neg\neg\varphi = \varphi$ for every φ in our special form.

Notation 6.2.2 *We use $\mathcal{G}$ for the generic filter, G for $\cup\{V(p)|p \in \mathcal{G}\}$, the function associated with $\mathcal{G}$ that we are building and $\mathtt{G}$ for the symbol in language that stands for that function.*

We now define the forcing relation $p \Vdash \varphi$.

deffor **Definition 6.2.3** *Given a notion of forcing $\mathcal{P}$, we define the relation p* forces *φ, $p \Vdash \varphi$, for $p \in P$ and sentences φ of our specified form, by induction on the number of quantifiers in φ.*

- If φ is a Σ_1 sentence $\exists s(\varphi_{1+k}(\mathtt{G}, e, e, \bar{c}, s)\downarrow)$ then $p \Vdash \varphi$ if and only if $\phi_{1+k}(V(p), e, e, \bar{c})\downarrow$.
- If φ is a Π_1 sentence $\forall s(\neg(\varphi_{1+k}(\mathtt{G}, e, e, \bar{c}, s)\downarrow))$ then $p \Vdash \varphi$ if and only if $\forall q \leq p(\neg\phi_{1+k}(V(q), e, e, \bar{c})\downarrow))$.
- If φ is a Σ_{n+1} formula $\exists x\theta(x)$ then $p \Vdash \varphi$ if and only if $\exists m(p \Vdash \theta(m))$.
- If φ is a Π_{n+1} formula $\forall x\psi(x)$ then $p \Vdash \varphi$ if and only if $\forall q \leq p\forall m(q \nVdash \neg\psi(m))$.

Remark 6.2.4 *We have restricted our definition of forcing to sentences of very specific forms and consider only these in formal proofs about properties of forcing. In applications, however, we abuse our notation by writing $p \Vdash \theta(\mathtt{G})$ for any sentence θ of arithmetic. Formally, we assume some uniform procedure to replace any θ by an equivalent sentence in our special form. In view of the relation between forcing (which depends on the syntactical form) and truth (which does not up to semantic equivalence) in Theorem 6.2.15, this will make no difference in terms of our generic sets satisfying the sentence θ.*

relfor **Remark 6.2.5 (Relativization)** *Just as we often relativized results in Chapter 5, we often want to consider arithmetic and forcing with extra function parameters $\bar{h}$ for fixed functions $\bar{h}$ depending on the circumstances. For example, in Exercise 5.1.6, we built $A_i \geq_T X$ with $A_0|_T A_1$ by relativizing the construction of Theorem 5.1.1 to X. So too, we now want to be able to fix some $\bar{h}$ and talk about the properties of G in a language that allows function parameters for these $\bar{h}$. We do this in the natural way. The basic language of arithmetic described in §4.4 allowed for multiple function parameters ?? So we have formulas $\theta(G, \bar{h}, \bar{c}, \bar{x})$ of arithmetic which are equivalent to ones of the form $\bar{Q}\bar{x}\exists s(\varphi_{1+k+l}(G, \bar{h}, e, e, \bar{c}, \bar{x}, s)\downarrow)$ or $\bar{Q}\bar{x}\forall s(\neg(\varphi_{1+k+l}(G, \bar{h}, e, e, \bar{c}, \bar{x}, s)\downarrow))$ where we similarly use the versions of the universal partial functions of §2.1 with multiple oracles. For the definition of forcing we then, of course, use the recursive in $\bar{h}$ relations $\varphi_{1+k+l}(\sigma, \bar{h}, e, e, \bar{c}, \bar{x})$. As usual, we typically leave such matters of adding parameters and relativizing to the reader.*

compfor **Theorem 6.2.6** *If $\mathcal{P}$ is a notion of forcing then, for $n \geq 1$, forcing for $\Sigma_n^{\mathcal{P}}$ ($\Pi_n^{\mathcal{P}}$) sentences φ (i.e. whether $p \Vdash \varphi$) is a Σ_n (Π_n) in $\mathcal{P}$ relation.*

Proof. We proceed by induction on n. If φ is Σ_1 or Π_1 then $p \Vdash \varphi$ is directly defined as a $\Sigma_1^{\mathcal{P}}$ or $\Pi_1^{\mathcal{P}}$ formula, respectively. (The point here is that the $\phi_{1+k}(\sigma, e, e, \bar{c})$ are uniformly recursive, P and $q \leq p$ are recursive in $\mathcal{P}$ and V is recursive on P.) For $n \geq 1$ the result follows by induction and our definition of forcing. ■

Exercise 6.2.7 *In fact, if φ is Σ_1 then $p \Vdash \varphi$ is a recursive in $\mathcal{P}$ relation.*

ext **Exercise 6.2.8** *If $p \Vdash \varphi$ and $q \leq p$ then $q \Vdash \varphi$.*

We now want to tackle the question of how much genericity do we need to make forcing equal truth for generic filters and functions in the sense that if $p \Vdash \varphi$, $p \in \mathcal{G}$ and $\mathcal{G}$ is sufficiently generic then $\varphi(G)$ holds and, in the other direction, if $\varphi(G)$ holds then there is a $p \in \mathcal{G}$ such that $p \Vdash \varphi$.

ngeneric **Definition 6.2.9** *Let $\mathcal{P}$ be a notion of forcing and $\mathcal{G}$ a filter on $\mathcal{P}$. For $n \geq 1$, We say $\mathcal{G}$ is n-*generic *for $\mathcal{P}$ if, for every Σ_n in $\mathcal{P}$ subset S of $\mathcal{P}$,*

$$\exists p \in \mathcal{G}(p \in S \ \vee \ \forall q \leq p(q \notin S)).$$

*We say that $\mathcal{G}$ is ω-*generic *(or simply* generic*) for $\mathcal{P}$ if it is n-generic for all n.*

*Similarly, the descending sequence $\langle p_n \rangle$ of conditions is n-*generic *for $\mathcal{P}$ if, for every Σ_n in $\mathcal{P}$ subset S of $\mathcal{P}$, there is an m such that $p_m \in S$ or $\forall q \leq p_m(q \notin S)$. The sequence is ω-* generic *(or simply* generic*) for $\mathcal{P}$ if it is n-generic for $\mathcal{P}$ for all n.*

*The function G determined by an (n-)generic filter or sequence is also said to be (n-)*generic*. A degree $\mathbf{g}$ is (n)-generic for $\mathcal{P}$ if there is a $G \in \mathbf{g}$ which is (n)-generic for $\mathcal{P}$.*

*These notions all relativize to an arbitrary h in the obvious way. We then say, for example, that G is n-*generic relative to (or over) h*. (We include h as a parameter in our languages for arithmetic and forcing as in Remark* relfor *6.2.5 and consider sets S which are Σ_n in $\mathcal{P} \oplus h$.)*

The following equivalence is now immediate.

enericdense **Proposition 6.2.10** *Let $\mathcal{C}_n$ be the class of dense sets $\{p : p \in S_e \ \vee \ \forall q \leq p(q \notin S_e)\} = D_{n,e}$ for all Σ_n in $\mathcal{P}$ subsets S_e of $\mathcal{P}$. A filter $\mathcal{G}$ (or a descending sequence $\langle p_n \rangle$) is n-generic iff $\mathcal{G}$ ($\langle p_n \rangle$) is $\mathcal{C}_n$-generic.*

enmeetdense **Exercise 6.2.11** *If $D \subseteq \mathcal{P}$ is dense and $\Sigma_n^{\mathcal{P}}$ then D meets every n-generic $\mathcal{G}$. If D is dense below p and $\Sigma_n^{\mathcal{P}}$ then D meets every n-generic $\mathcal{G}$ containing p.*

We now point out an important sense in which the variants on Cohen forcing mentioned in Example cohenfor 6.1.2 are equivalent.

cohenvar **Exercise 6.2.12** *Let $\mathcal{P}_k$ be $(k^{<\omega}, \supseteq)$ and $\mathcal{P}_\omega$ be $(\mathbb{N}^{<\omega}, \supseteq)$. (So $\mathcal{P}_2$ is Cohen forcing). Show that for every n a degree $\mathbf{g}$ is $\mathcal{P}_\omega$ (n-)generic if and only if it is $\mathcal{P}_k$ (n-)generic for every $k \in \mathbb{N}$ iff and only if it is $\mathcal{P}_k$ (n-)generic for some $k \in \mathbb{N}$.*

Hint: Consider, for example, the map that first takes any $f \in 2^{\mathbb{N}}$ to $h : \mathbb{N} \to \mathbb{N}$ given by writing f as $0^{m_0}1^{m_1}0^{m_2}\ldots$ where each m_i is chosen maximal and we are taking 0^0 to be the empty string. Then we set $h(i) = m_i$. Prove that f is Cohen (n-)generic if and only if h is $\mathcal{P}_\omega$ (n-)generic and in all these cases $f \equiv_T h$.

To build an n-generic $\mathcal{G}$ we proceed as in the construction of a generic for an arbitrary countable class of dense sets. We can now also calculate how hard it is to carry out this construction.

ngenericOn **Proposition 6.2.13** *For any notion of forcing $\mathcal{P}$ and each $n \geq 1$, there is an n-generic sequence $\langle p_k \rangle \leq_T \mathcal{P}^{(n)}$ and so its associated n-generic G is also recursive in $\mathcal{P}^{(n)}$. There is also a generic sequence $\langle p_k \rangle$ such that it and its associated G is recursive in $\mathcal{P}^{(\omega)}$. Moreover, for any $p \in P$ we may require that $p_0 = p$ and so $V(p) \subseteq G$.*

Proof. Fix n. We build a generic sequence $\langle p_n \rangle$ for the $\mathcal{C}_n$ of Proposition ngenericdense 6.2.10 recursively in $\mathcal{P}^{(n)}$. We begin with p_0 the given $p \in P$. If we have already defined p_s we find, recursively in $\mathcal{P}^{(n)}$, a $q \leq_{\mathcal{P}} p_s$ which is in $D_{n,s+1}$. This procedure clearly constructs the desired sequence and is recursive in $\mathcal{P}^{(n)}$ by definition of the $D_{n,e}$. For ω-genericity we simply carry out this construction for the collection $\{D_{n,e} | n, e \in \omega)\}$ recursively in $\mathcal{P}^{(\omega)}$. That G is recursive in $\mathcal{P}^{(n)}(\mathcal{P}^{(\omega)})$ follows from Proposition assocgen 6.1.23. ∎

decide **Definition 6.2.14** *A condition p* decides a sentence φ *if $p \Vdash \varphi$ or $p \Vdash \neg\varphi$.*

for=t **Theorem 6.2.15** *If $\mathcal{G}$ is n-generic for a notion of forcing $\mathcal{P}$ and $\varphi(\mathtt{G}) \in \Sigma_k$ or Π_k, $k \leq n$, then there is a $p \in \mathcal{G}$ which decides $\varphi(\mathtt{G})$. Moreover, if there is a $p \in \mathcal{G}$ such that $p \Vdash \varphi(\mathtt{G})$ then $\varphi(G)$ holds.*

Proof. For the first assertion, note that the case for Π_k sentences φ follows from that for Σ_k sentences since $\neg\varphi$ is Σ_k and $\neg\neg\varphi$ is φ. Suppose then that φ is Σ_k and consider the set $S_\varphi = \{p | p \Vdash \varphi\}$. By Theorem compfor 6.2.6, S_φ is Σ_k and so by the definition of n-genericity, there is a $p \in \mathcal{G}$ such that $p \Vdash \varphi$ or $\forall q \leq p(q \nVdash \varphi)$. If $p \Vdash \varphi$ we are done so suppose $\forall q \leq p(q \nVdash \varphi)$.

We now want to prove that $p \Vdash \neg\varphi$. If $k = 1$, i.e. φ is of the form $\exists s(\varphi_{1+k}(\mathtt{G}, e, e, \bar{c}, s) \downarrow$), this follows immediately from the definitions of forcing a Σ_1 sentence and the Π_1 sentence which is its negation: $q \nVdash \varphi \Leftrightarrow \neg\phi_{1+k}(V(q), e, e, \bar{c}) \downarrow$ and so by our assumption $\forall q \leq p(\neg\phi_{1+k}(V(q), e, e, \bar{c}) \downarrow)$, i.e. $p \Vdash \forall s(\neg(\varphi_{1+k}(\mathtt{G}, e, e, \bar{c}, s) \downarrow))$ while $\neg\varphi = \forall s(\neg(\varphi_{1+k}(\mathtt{G}, e, e, \bar{c}, s) \downarrow))$.

If $k > 1$ then $\varphi = \exists x \psi(x)$ for some Π_{k-1} sentences ψ. By the definition of forcing for Σ_k sentences, we have that $\forall q \leq p \forall m(q \nVdash \psi(m))$. Keeping in mind that $\neg\neg\psi = \psi$, this is precisely the definition of $p \Vdash \forall x \neg\psi$. As $\forall x \neg\psi = \neg\varphi$, we have the desired result.

We now prove the second assertion of the theorem by induction on k. Suppose φ is Σ_1 and so of the form $\exists s(\varphi_{1+k}(\mathtt{G}, e, e, \bar{c}, s) \downarrow)$. As $p \Vdash \varphi$, $\phi_{1+k}(V(p), e, e, \bar{c}) \downarrow$. As $V(p) \subset G$, $\varphi_{1+k}(G, e, e, \bar{c}, |V(p)|) \downarrow$ and so $\exists s(\varphi_{1+k}(G, e, e, \bar{c}, s) \downarrow)$ as required.

Next suppose that φ is Π_1 and so of the form $\forall s(\neg(\varphi_{1+k}(\mathtt{G}, e, e, \bar{c}, s) \downarrow))$, $p \Vdash \varphi$ and, for the sake of a contradiction, that $\forall s(\neg(\varphi_{1+k}(G, e, e, \bar{c}, s) \downarrow))$ does not hold. This means that there is some s such that $\varphi_{1+k}(G, e, e, \bar{c}, s) \downarrow$ and so $\phi_{1+k}(G \restriction s, e, e, \bar{c}) \downarrow$. Now there is an $r \in V_s \cap \mathcal{G} \neq \emptyset$ (by Definition forcing2 6.1.12) and so a $q \geq p, r$ in $\mathcal{G}$ by Definition filter 6.1.7. As $V(q) \supseteq V(p)$ (again by Definition forcing2 6.1.12), $\phi_{1+k}(V(q), e, e, \bar{c}) \downarrow$ (by the use property of codeTM §2.1). As $q \leq p$ we have contradicted the assumption that $p \Vdash \varphi$ as desired.

Next, we consider Σ_{k+1} and Π_{k+1} sentences φ for $k \geq 1$. If $\varphi = \exists x \psi(\mathtt{G}, x)$ with ψ a Π_k sentence and $p \Vdash \varphi$, then by definition $p \Vdash \psi(\mathtt{G}, m)$ for some m. By induction, $\psi(G, m)$ holds and then so does $\varphi = \exists x \psi(G, x)$ as required. Finally, suppose, for the sake of a contradiction, that $\varphi = \forall x \psi(\mathtt{G}, x)$ with ψ a Σ_k sentence, $p \Vdash \varphi$ but $\forall x \psi(G, x)$ does not hold. As $\forall x \psi(G, x)$ fails, there is some m such that $\neg\psi(G, m)$ holds. The first part of our Theorem supplies an $r \in \mathcal{G}$ such that $r \Vdash \psi(G, m)$ or $r \Vdash \neg\psi(\mathtt{G}, m)$. The first alternative contradicts our inductive assumption as ψ is Σ_k. Thus $r \Vdash \neg\psi(\mathtt{G}, m)$. Once more by Definition 6.1.7, we have a $q \leq p, r$. As $q \Vdash \neg\psi(\mathtt{G}, m)$, we have contradicted the definition of $p \Vdash \forall x \psi(\mathtt{G}, x)$ as desired. ■

for=tseq **Exercise 6.2.16** *Prove the analog of Theorem 6.2.15 for n-generic sequences $\langle p_i \rangle$: For any Σ_k or Π_k formula $\varphi(\mathtt{G})$ there is an i such that $p_i \Vdash \varphi$ or $p_i \Vdash \neg\varphi$. Moreover, if, for any i, $p_i \Vdash \varphi(\mathtt{G})$ then $\varphi(G)$ holds.*

We now analyze the degree theoretic properties of sets with various amounts of genericity. We begin with some connections between genericity and lowness. The basic Proposition provides two corollaries. The first improves Proposition 6.2.13. The second is specific to notions of forcing similar to that of Cohen.

enericlowng **Proposition 6.2.17** *For any notion of forcing $\mathcal{P}$ and each $n \geq 1$ and n-generic filter or sequence Z with associated n-generic G, $(\mathcal{P} \oplus G)^{(n)} \leq_T Z \oplus \mathcal{P}^{(n)}$. Similarly, if Z is generic then $(\mathcal{P} \oplus G)^{(\omega)} \leq_T Z \oplus \mathcal{P}^{(\omega)}$ for the associated G.*

Proof. Consider the case for n-generics. As the question of whether $e \in (\mathcal{P} \oplus G)^{(n)}$ is uniformly Σ_n in $\mathcal{P} \oplus G$, we can recursively find the $\Sigma_n^{\mathcal{P}}$ formula $\theta_e(\mathtt{G})$ of our forcing language such that $\theta_e(G) \Leftrightarrow e \in G^{(n)}$. Now as $p \Vdash \theta_e(\mathtt{G})$ and $p \Vdash \neg\theta_e(\mathtt{G})$ are recursive in $\mathcal{P}^{(n)}$ (Theorem 6.2.6) and there is a p in Z that decides $\theta_e(\mathtt{G})$ by Theorem 6.2.15 or Exercise 6.2.16, we can find one recursively $Z \oplus \mathcal{P}^{(n)}$. Theorem 6.2.15 or Exercise 6.2.16 then tells us that $e \in (\mathcal{P} \oplus G)^{(n)} \Leftrightarrow p \Vdash \theta_e(\mathtt{G})$. The proof for full genericity is essentially the same using $\mathcal{P}^{(\omega)}$. ■

genericlown **Corollary 6.2.18** *For any notion of forcing $\mathcal{P}$ and each $n \geq 1$ there is an n-generic G such that $G^{(n)} \leq_T \mathcal{P}^{(n)}$. There is also a generic G such that $G^{(\omega)} \leq_T \mathcal{P}^{(\omega)}$.*

Proof. By Proposition 6.2.13, there is an n-generic sequence $\langle p_i \rangle$ recursive in $\mathcal{P}^{(n)}$ and a generic one recursive in $\mathcal{P}^{(\omega)}$. Now apply Proposition 6.2.17. ■

cohenngen **Corollary 6.2.19** *If G is n-generic for Cohen forcing then $G^{(n)} \equiv_T G \vee 0^{(n)}$. Similarly, if G is generic for Cohen forcing, $G^{(\omega)} \equiv_T G \vee 0^{(\omega)}$.*

Proof. For Cohen forcing the filter $\mathcal{G}$ with which G is associated is just $\{\sigma | \sigma \subset G\}$ which is clearly recursive in G. Now apply Proposition 6.2.17. ■

We can now generalize the Friedberg Completeness Theorem 5.3.1 to iterations of the jump. The proofs of Exercises 6.2.20 and 6.2.21 combine the ideas in the proofs of Proposition 6.2.20 and Theorem 5.3.1.

genFrcomp **Exercise 6.2.20** *For every $C \geq_T 0^{(n)}$ there is a Cohen n-generic G such that $C \equiv_T 0^{(n)} \oplus G \equiv_T G^{(n)}$. The same is true if we replace n by ω.*

More generally, we have the following.

genFrcomp2 **Exercise 6.2.21** *If for every $p \in \mathcal{P}$ there are $q, r \leq p$ such that $V(q)|V(r)$ then, for every $C \geq_T \mathcal{P}^{(n)}$ we can find an n-generic G such that $C \equiv_T \mathcal{P}^{(n)} \oplus G \equiv_T (\mathcal{P} \oplus G)^{(n)} \equiv_T \mathcal{P}^{(n)} \oplus G^{(n)}$. The same is true if we replace n by ω.*

Exercise 6.2.22 *Find an A-recursive notion of forcing for which the analog of Corollary cohenngen 6.2.19 does not hold, i.e. there is an n-generic G with $G^{(n)} \not\leq_T G \vee A^{(n)}$. ??Hint??*

The next proposition gives almost all of the incomparability and embeddability results for countable sets of Chapter embeddings 5 in one fell swoop.

genindep **Proposition 6.2.23** *If $\mathcal{G}$ is Cohen 1-generic then the columns $G^{[i]} = \{x|\langle i, x\rangle \in G\}$ of G form a very independent set, i.e. $\forall j(G^{[j]} \not\leq_T G^{[\hat\jmath]})$.*

Proof. Note that we are using the column notation for sets and binary strings from Notation colnot 5.1.13. For each e and j we want to show that $\Phi_e^{G^{[\hat\jmath]}} \neq G^{[j]}$. Consider the Σ_1 sets $S_{e,j} = \{p : p \Vdash \exists x(\Phi_e^{\mathtt{G}^{[\hat\jmath]}}(x) \downarrow\neq \mathtt{G}^{[j]}(x))\}$. As G is 1-generic, there is $p \subset G$ such that $p \in S_{e,j}$ or $(\forall q \leq p)(q \notin S_{e,j})$. (The 1-generic filter with which G is associated is $\mathcal{G} = \{p|p \subset G\}$.) In the first case, $\exists x(\Phi_e^{G^{[j]}}(x) \downarrow\neq G^{[\hat\jmath]}(x))$ (by Theorem for=t 6.2.15) and so $\Phi_e^{G^{[\hat\jmath]}} \neq G^{[j]}$ as required. In the second case, we claim that $\Phi_e^{G^{[\hat\jmath]}}$ is not total. If it were, let $\langle j, x\rangle$ be outside the domain of p. We must then have some $q \subset G$ with $q \leq p$, $q(\langle j, x\rangle)\downarrow$ and $\Phi_e^{q^{[j]}}(x)\downarrow$. Now let $\hat q(\langle j, x\rangle) = 1 - q(\langle j, x\rangle)$ and $\hat q(z) = q(z)$ for $z \neq \langle j, x\rangle$. So $\hat q^{[\hat\jmath]} = q^{[\hat\jmath]}$ and $\Phi_e^{q^{[\hat\jmath]}}(x)\downarrow= \Phi_e^{\hat q^{[\hat\jmath]}}(x)\downarrow$ but $\hat q(\langle j, x\rangle) \neq q(\langle j, x\rangle$. Thus (by the definition of forcing for Σ_1 sentences) one of q and $\hat q$ (both of which extend p) is in S_e for the desired contradiction. ■

Exercise 6.2.24 *The Theorems and Propositions of this section relativize to an arbitrary X. For example, Proposition genindep 6.2.23 now says that if G is 1-generic over X, then the independence results hold even relative to X, i.e. $\forall j(G^{[j]} \not\leq_T X \oplus G^{[\hat\jmath]})$.*

Exercise 6.2.25 *If G is Cohen 1-generic over X and $A, B \leq_T X$ then*

$$A \leq_T B \Leftrightarrow A \oplus G \leq_T B \oplus G.$$

In particular, if $X >_T 0$, $G \mid_T X$.

Exercise 6.2.26 *If G is Cohen 1-generic over $X >_T 0$, then $G \wedge X = 0$, i.e. G and X form a minimal pair.*

Exercise 6.2.27 *Prove that if G is Cohen n-generic then the $G^{[i]}$ are very mutually Cohen n-generic in the sense that each $G^{[i]}$ is Cohen n-generic over $G^{[\hat\imath]}$.*

Exercise 6.2.28 *Translate the Exact Pair Theorem into the language of forcing. Hint: Given $\langle C_i \rangle$, define a notion of forcing $\mathcal{P}$ with conditions $\langle \alpha, \beta, n \rangle$ for $\alpha, \beta \in 2^{<\omega}$ and $n \in \mathbb{N}$. The ordering is given by $\langle \alpha', \beta', n' \rangle \leq \langle \alpha, \beta, n \rangle$ if $\alpha' \supseteq \alpha$, $\beta' \supseteq \beta$, $n' \geq n$ and, for $i < n$, if $\alpha'(\langle i, x \rangle) \downarrow$ but $\alpha(\langle i, x \rangle) \uparrow$ then $\alpha'(\langle i, x \rangle) = C_i(x)$ and similarly for β' and β.*

Exercise 6.2.29 *Let $f : \mathbb{N} \to \{0, 1, 2\}$ and $\{d_n\}$ list the x such that $f(x) = 2$ in increasing order. For any $A \in 2^{\mathbb{N}}$, we let $f_A(x) = A(n)$ if $x = d_n$ for some n and $f_A(x) = f(x)$ otherwise. Construct an f such that f_A is Cohen 1-generic for every $A \in 2^{\mathbb{N}}$. Hint: make f 1-generic for conditions $p \in \{0, 1, 2\}^{<\omega}$.*

Exercise 6.2.30 *Show that the Cohen 1-generic degrees generate $\mathcal{D}$. Hint: Fix an $h \in 2^{\mathbb{N}}$. Make the f of the previous construction 1-generic relative to A. Show that for any $j \neq k$, $(f_A^{[j]} \oplus f_{\bar{A}}^{[j]}) \wedge (f_A^{[k]} \oplus f_{\bar{A}}^{[k]}) \equiv_T A$.*

??Probably write out this proof as Proposition not exercise.??

We close this section with a slight variation of our previous constructions that is needed in §def<0' 8.4.

nfortwosets **Proposition 6.2.31** *If $\mathcal{P}$ is a recursive notion of forcing and C_0 and C_1 are low sets, i.e. $C_0' \equiv_T 0' \equiv_T C_1'$ then there is a G which is 1-generic for $\mathcal{P}$ over C_0 and over C_1 so that, in particular, both $G \oplus C_0$ and $G \oplus C_1$ are low.*

Proof. Build a generic sequence meeting the dense sets $\{p : p \in S_{e,i} \ \vee \ \forall q \leq p(q \notin S_{e,i})\} = D_{e,i}$ for all Σ_n in C_i subsets $S_{e,i}$ of $\mathcal{P}$ for $i \in \{0, 1\}$ as in the proof of Proposition ngenericOn 6.2.13. As both C_i are low, $0'$ can uniformly compute extensions in each $D_{e,i}$ of any p so the generic sequence and the associated G are recursive in $0'$. Moreover, when we meet $D_{e,i}$ we also decide which clause in the definition is satisfied. To decide if $j \in (G \oplus C_i)'$, i.e. if $\Phi_j^{G \oplus C_i}(j) \downarrow$ we recursively find the e such that $S_{e,i} = \{p | p \Vdash \Phi_j^{G \oplus C_i}(j) \downarrow\}$. Then, as in the proof of Proposition ngenericlowng 6.2.17, $j \in (G \oplus C_i)'$ if and only if at the stage we met $D_{e,i}$ we met $S_{e,i}$. ■

Notes: Forcing in arithmetic was introduced in Feferman [1965]. It has since been used in various formulations by many people. Hinman [1969] introduced a version of n-genericity. Two important early papers applying forcing to degree theory are Jockusch and Posner [1978] and Jockusch [1980] in which many of the results of this section appear for the special but typical case of Cohen forcing. A systematic development of degree theory based on forcing was first presented in Lerman [1983]. Our approach attempts to both simplify and generalize previous versions. A similar version is in Cai and Shore [2012].

Give specific references for specific theorems and exercises??

latembsec 6.3 Embedding Lattices

We have so far studied questions of embedding countable partial orders (and usl's) in $\mathcal{D}$ which is itself an usl. Now we know that $\mathcal{D}$ is not a lattice (Corollary notlattice 5.2.18) but we also know that some pairs of degrees do have greatest lower bounds in $\mathcal{D}$ (Theorem minpair 5.2.9). Thus we can ask which lattices can be embedded in $\mathcal{D}$ preserving the full lattice structure. We now prove that every countable lattice can be embedded in $\mathcal{D}$.

latemb **Theorem 6.3.1 (Lattice Embedding Theorem)** *Every countable lattice $\mathcal{L}$ is embeddable in $\mathcal{D}$ preserving the lattice structure.*

For later convenience, we actually want to prove an *a priori* stronger statement about partial lattices.

Definition 6.3.2 *A* partial lattice *$\mathcal{L}$ is a partial order $\leq_{\mathcal{L}}$ on its domain L together with partial functions $\wedge$(meet) and $\vee$ (join) which satisfy the usual definitions when defined, i.e. if $x \wedge y = z$ then z is the greatest lower bound of x and y in $\leq_{\mathcal{L}}$; if $x \vee y = z$ then z is the least upper bound of x and y in $\leq_{\mathcal{L}}$. We say that $\mathcal{L}$ is recursive (in A) if L and $\leq_{\mathcal{L}}$ are recursive (in A) and $\vee$ and $\wedge$ are partial recursive (in A) functions on L.*

Now, actually every partial lattice can be embedded into a lattice.

extparlat **Theorem 6.3.3** *If $\mathcal{L}$ is a partial lattice with least element 0 and greatest element 1 then there is a lattice $\hat{\mathcal{L}}$ and an embedding $f : \mathcal{L} \to \hat{\mathcal{L}}$ which preserves 0, 1, order and all meets and joins that are defined in $\dot{\mathcal{L}}$.*

Proof. Consider the lattice $\mathcal{I}$ of nonempty ideals of $\mathcal{L}$, i.e. nonempty subsets I of L closed downward and under join in $\mathcal{L}$ (when defined). The ordering on $\mathcal{I}$ is given by set inclusion. Meet is set intersection and the join of I_1 and I_2 is the smallest ideal containing both of them. The map that sends $x \in \mathcal{L}$ to $I_x = \{y \in L | y \leq_{\mathcal{L}} x\}$, the principle ideal generated by x, is easily seen to be the desired embedding into the sublattice $\hat{\mathcal{L}}$ of $\mathcal{I}$ generated by the principle ideals. ■

ingenparlat **Corollary 6.3.4** *If $\mathcal{L}$ is finitely generated as a partial lattice, then the $\hat{\mathcal{L}}$ of Theorem extparlat 6.3.3 may be taken to be finitely generated as a lattice, indeed it is generated by the images of the generators of $\mathcal{L}$ under f.*

Proof. Consider the $\hat{\mathcal{L}}$ provided by Theorem extparlat 6.3.3. The sublattice $\mathcal{L}'$ of $\hat{\mathcal{L}}$ generated (as a lattice) by the image under f of the generators of $\mathcal{L}$ is a finitely generated lattice into which f also gives an embedding. ■

Extending partial lattices to lattices is not necessarily effective.

paul **Exercise 6.3.5** *There is a recursive partial lattice $\mathcal{L}$ which cannot be recursively embedded in a recursive lattice. Hint:??(Paul Shafer).??*

Exercise 6.3.6 *??Put in effective embeddings of p.o. and usl... all the way into Boolean algebras where do p.o. and usl embeddings?? Move to appendix??*

Thus as far as a simple embedding theorem is concerned, it may seem that there is no reason to use partial lattices but both effectiveness considerations and convenience come into play. It is certainly often more convenient to specify a partial lattice than to decide all the meets and joins. Thus we state our theorem for partial lattices.

parlatemb **Theorem 6.3.7 (Partial Lattice Embedding)** *If $\mathcal{L}$ is a partial lattice recursive in A with least element 0 and greatest element 1 then there is an embedding $f : \mathcal{L} \to \mathcal{D}$ which preserves order and all meets and joins that are defined in $\mathcal{L}$. Moreover, we may guarantee that $f(0) = \mathbf{a} \leq_T f(1) \leq_T \mathbf{a}''$ and that, for $x \in \mathcal{L}$, $f(x)$ is uniformly recursive in $f(1)$, in the sense that we have sets G_x of degree $f(x)$ which are uniformly recursive in $f(1)$.*

To prove Theorem parlatemb 6.3.7, we need some lattice theory. In particular, we use a type of lattice representations called lattice tables.

latrep **Definition 6.3.8** *A* lattice table *for the partial lattice $\mathcal{L}$ is a collection, Θ, of maps $\alpha : L \to \mathbb{N}$ such that for every $x, y \in L$ and $\alpha, \beta \in \Theta$*

1. *$\alpha(0) = 0$.*
2. *If $x \leq_{\mathcal{L}} y$ and $\alpha(y) = \beta(y)$ then $\alpha(x) = \beta(x)$.*
3. *If $x \not\leq_{\mathcal{L}} y$ then there are $\alpha, \beta \in \Theta$ such that $\alpha(y) = \beta(y)$ but $\alpha(x) \neq \beta(x)$.*
4. *If $x \vee y = z$, $\alpha(x) = \beta(x)$ and $\alpha(y) = \beta(y)$ then $\alpha(z) = \beta(z)$.*
5. *If $x \wedge y = z$ and $\alpha(z) = \beta(z)$ then there are $\gamma_1, \gamma_2, \gamma_3 \in \Theta$ such that $\alpha(x) = \gamma_1(x)$, $\gamma_1(y) = \gamma_2(y)$, $\gamma_2(x) = \gamma_3(x)$, $\gamma_3(y) = \beta(y)$. Such γ_i are called* interpolants for α and β (with respect to x, y and z).

The representation of lattices by lattice tables is closely related to the more standard (in lattice theory) representation by equivalence relations.

eqrels **Definition 6.3.9** *We define equivalence relations on Θ for each $x \in \mathcal{L}$ by $\alpha \equiv_x \beta$ if and only if $\alpha(x) = \beta(x)$. For sequences p, q from Θ of length n and $x \in L$, we say $p \equiv_x q$ if $p(k) \equiv_x q(k)$ for every $k < n$. We also write, for example, $p \equiv_{x,y} q$ to mean that $p \equiv_x q$ and $p \equiv_y q$. For $g \in \Theta^\omega$, we write $p \equiv_x g$ or $p \equiv_{x,y} g$ to mean that $p \equiv_x g \upharpoonright |p|$ or $p \equiv_{x,y} g \upharpoonright |p|$, respectively.*

ordeqrel **Definition 6.3.10** *For arbitrary equivalences relation E, $\hat{E}$ on a set S, we say E is* larger *or* coarser *than $\hat{E}$ if $(\forall a, b \in S)(a \equiv_{\hat{E}} b \Rightarrow a \equiv_E b)$. Similarly, E is* finer *or* smaller *than $\hat{E}$ if $(\forall a, b \in S)(a \equiv_E b \Rightarrow a \equiv_{\hat{E}} b)$.*

lateqrel **Remark 6.3.11** *With this ordering on equivalence relations (on S), the join of E and $\hat{E}$ is simply their intersection. Their meet is the smallest equivalence class on S that contains their union. This is also the transitive closure of their union under the two relations.*

The conditions of Definition latrep 6.3.8 can now be restated in terms of these equivalence relations:

1. $\alpha \equiv_0 \beta$ for all α and β and so $\equiv_0$ is the largest congruence class, i.e. the one identifying all elements. At the other end, $\equiv_1$is the smallest congruence relation, indeed, it agrees with equality: By Definition latrep 6.3.8(1), $\alpha(1)$ uniquely determines each $\alpha \in \Theta$.

2. If $x \leq y$ then $\alpha \equiv_y \beta$ implies $\alpha \equiv_x \beta$ for all α and β and so $\equiv_x$ is larger than $\equiv_y$.

3. If $x \nleq_{\mathcal{L}} y$ then there are α and β such that $\alpha \equiv_y \beta$ but $\alpha \not\equiv_x \beta$ and so $\equiv_x$ is not larger than $\equiv_y$.

4. If $x \vee y = z$ and $\alpha \equiv_x \beta$ and $\alpha \equiv_y \beta$ then $\alpha \equiv_z \beta$ and so $\equiv_z$ is the meet of $\equiv_x$ and $\equiv_y$.

5. If $x \wedge y = z$ then there are $\gamma_1, \gamma_2, \gamma_3 \in \Theta$ such that $\alpha \equiv_x \gamma_1 \equiv_y \gamma_2 \equiv_x \gamma_3 \equiv_y \beta$. So $\equiv_z$ is certainly contained in the join of $\equiv_x$ and $\equiv_y$. It is part of the theorem that we can arrange it so that chains of length three suffice to generate the entire transitive closure.

Thus a lattice table Θ produces a representation by equivalence relations with the dual ordering. A reason for reversing the order is that $\mathcal{D}$ is only an uppersemilattice. So joins always exist and we want them to correspond to the simple operation on equivalence relations of intersection. On the other hand, meets do not always exist and they then correspond to join on equivalence relations which requires work to construct.

We now prove our representation theorem in terms of lattice tables.

repthm1 **Theorem 6.3.12 (Representation Theorem)** *If $\mathcal{L}$ is a recursive (in A) partial lattice with $0, 1$ then there is a uniformly recursive (in A) lattice table Θ for $\mathcal{L}$.*

Proof. Define $\beta_{x,i}$ for $x, y \in L$, $i = 0, 1$ by

$$\beta_{x,0}(y) = \begin{cases} \langle x, 0 \rangle \text{ if } y \neq 0 \\ 0 \text{ if } y = 0 \end{cases} \qquad \beta_{x,1}(y) = \begin{cases} \beta_{x,0}(y) \text{ if } y \leq_{\mathcal{L}} x \\ \langle x, 1 \rangle \text{ if } y \nleq_{\mathcal{L}} x \end{cases}$$

The set of these $\beta_{x,i}$ satisfy (1), (2), (3) and (4). We now want to sequentially close off under adding interpolants as required in (5) for each relevant instance. To do so, we have some dovetailing procedure which does the following. Consider $x \wedge y = z$ and $\alpha \equiv_z \beta$. We want to add $\gamma_1, \gamma_2, \gamma_3$ as required in (5) and preserve the truth of (1)-(4) in the expanded set. If $x \leq_{\mathcal{L}} y$ or $y \leq_{\mathcal{L}} x$, it is easy to do so just using α and β. If not (i.e. $x \nleq_{\mathcal{L}} y$ and $y \nleq_{\mathcal{L}} x$), then choose new numbers a, b, c, d not used yet and for $w \in L$ let

$$\gamma_1(w) = \begin{cases} \alpha(s) \text{ if } w \leq_{\mathcal{L}} x \\ a \text{ if } w \nleq_{\mathcal{L}} x \end{cases} \qquad \gamma_2(w) = \begin{cases} \gamma_1(w) \text{ if } w \leq_{\mathcal{L}} y \\ b \text{ if } w \leq_{\mathcal{L}} x \text{ and } w \nleq_{\mathcal{L}} y \\ c \text{ otherwise} \end{cases} \qquad \gamma_3(w) = \begin{cases} \beta(w) \text{ if } w \leq_{\mathcal{L}} y \\ a \text{ if } w \leq_{\mathcal{L}} x \text{ and } w \nleq_{\mathcal{L}} y \\ d \text{ otherwise} \end{cases}$$

This is a recursive (in A) procedure and it is an Exercise to check that it works. ■

Exercise 6.3.13 *The construction given above provides a lattice table for $\mathcal{L}$.*

We now turn to the proof of our embedding theorem for partial latices.

Proof (of Theorem parlatemb 6.3.7). We assume that A and so $\mathcal{L}$ are recursive to simplify the notation. (Otherwise we just need to add on A to all our sets and procedures. We do this explicitly once or twice as a reminder.) We begin, then, with a recursive lattice table Θ for $\mathcal{L}$. We define a notion of forcing $\mathcal{P}$ with elements $p \in \Theta^{<\omega}$, the natural ordering $p \leq_{\mathcal{P}} q$ if $p \supseteq q$ and $V(p) = p$. Our generics are then maps $G : \mathbb{N} \to L$. Define, for $x \in L$, $G_x : \mathbb{N} \to \mathbb{N}$ by $G_x(n) = G(n)(x)$. The desired embedding f is given by $x \mapsto \deg(G_x)$ (or, in general, $x \mapsto \deg(G_x) \vee \mathbf{a}$).

We show that all the requirements for this map to be an embedding except for the preservation of $\wedge$ are satisfied if G is 1-generic. Our proof here of the preservation of $\wedge$ uses 2-genericity of G. We follow the numbering of clauses in Definition latrep 6.3.8.

1. By definition, 0 is preserved by our embedding as $G_0(n) = 0$ for all n and so $f(0) = \mathbf{0}$. As for $f(1)$, note that $G_1 \equiv_T G$: Given n, $G_1(n) = G(n)(1)$ so $G_1 \leq_T G(\oplus A)$. For the other direction, given $G_1(n)$ we already pointed out that there is only one $\alpha \in \Theta$ such that $\alpha(1) = G_1(n)$ and we can find this α, which is by definition $G(n)$, recursively. Thus $f(1) = \mathbf{g}$.

2. Suppose $x \leq_{\mathcal{L}} y$. We must show that $G_x \leq_T G_y$. Given n, we want to compute $G_x(n) = G(n)(x)$. Find any $\alpha \in \Theta$ such that $\alpha(y) = G_y(n)$. One exists because $G(n)$ is one such. As Θ is uniformly recursive we can search for and find one recursively in G_y. Then since $x \leq_{\mathcal{L}} y$ and $G(n) \equiv_y \alpha$, by Definition latrep 6.3.8(2) we have that $G(n) \equiv_x \alpha$ so $G(n)(x) = \alpha(x) = G_x(n)$.

4 Suppose $x \vee y = z$. We must show that $G_z \equiv_T G_x \oplus G_y$. By the preservation of order, $G_z \geq_T G_x \oplus G_y$, so it suffices to compute $G_z(n) = G(n)(z)$ from $G_x(n)$ and $G_y(n)$. We search for an $\alpha \in \Theta$ such that $\alpha(x) = G(n)(x)$ and $\alpha(y) = G(n)(y)$, i.e. $\alpha \equiv_{x,y} G(n)$. There is one and we can find it recursively in $G_x \oplus G_y$ as above. Now as $\alpha \equiv_{x,y} G(n)$, $\alpha \equiv_z G(n)$ by Definition latrep 6.3.8(3), so $\alpha(z) = G(n)(z)$.

Until this point, we have not actually used any genericity beyond the requirements imposed by Definition forcing2 6.1.12. We now turn to nonorder and meet.

3 Suppose $x \not\leq y$. We want to prove that $\Phi_e^{G_y} \neq G_x$ for every e. Suppose now that G is 1-generic (over A) and consider the Σ_1 sets

$$S_e = \{p \in \Theta^{<\omega} : p \Vdash (\exists n)[\Phi_e^{\mathtt{G}_y}(n) \downarrow \neq \mathtt{G}_x(n)]\}$$

The 1-genericity of G implies that there is a $p \in \mathcal{G} \cap S_e$ or there is a $p \in \mathcal{G}$ no extension of which is in S_e. Suppose $p \in \mathcal{G} \cap S_e$, then $\Phi_e^{G_y}(n) \neq G_x(n)$ and we are done. Otherwise, no extension of p is in S_e. Suppose then, for the sake of a contradiction, that $\Phi_e^{G_y} = G_x$. Let α and β be as in Definition latrep 6.3.8(3) for x and y. As the recursive sets $D_n = \{p|\exists m > n(p(m) = \alpha\}$ are clearly dense below p, the 1-genericity of $\mathcal{G}$ guarantees that there is a $q \leq p$ and an $m > |p|$ with $q(m) = \alpha$ and $q \in \mathcal{G}$ (Exercise ngenmeetdense 6.2.11). Moreover as $\Phi_e^{G_y}(m) \downarrow$, we may choose q so that $q \Vdash \Phi_e^{\mathtt{G}_y}(m) \downarrow$. Consider now the condition $\hat{q}$ such that $\hat{q}(k) = q(k)$ for $k \neq m$ and $\hat{q}(m) = \beta$. Our choice of α, β and q guarantees that $\hat{q} \leq p$, $q \equiv_y \hat{q}$ and $q \not\equiv_x \hat{q}$. Thus $q \Vdash \Phi_e^{\mathtt{G}_y}(m) \downarrow = i$ and $\hat{q} \Vdash \Phi_e^{\mathtt{G}_y}(m) \downarrow = i$ for some i but $q_x(m) \neq \hat{q}_x(m)$. So one of q and $\hat{q}$ is in S_e by definition for the desired contradiction.

5 Suppose that $x \wedge y = z$ and $\Phi_e^{G_x} = \Phi_e^{G_y} = D$. We want to prove that $D \leq_T G_z$. Now the assertion that $\Phi_e^{G_x}$ and $\Phi_e^{G_y}$ are total and equal is Π_2. By the 2-genericity (over A) of $\mathcal{G}$ and Theorem for=t 6.2.15, there is a $p \in \mathcal{G}$ such that p forces this sentence. Thus for each n and $q \leq p$, there is an $r \leq q$ and an i such that $r \Vdash \Phi_e^{\mathtt{G}_x}(n) = i = \Phi_e^{\mathtt{G}_y}(n)$. We now wish to compute $D(n)$ from G_z. As above, we can recursively in G_z find a $q \leq p$ and an i such that $q \Vdash \Phi_e^{\mathtt{G}_x}(n) = i = \Phi_e^{\mathtt{G}_y}(n)$ and $q_z \subset G_z$ (since some initial segment of G does this with $i = \Phi_e^{G_x}(n) = D(n)$).

We claim that $i = D(n)$. To see this consider a $t \in \mathcal{G}$ such that $t \leq p$, $|t| \geq |q|$ and $t \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = \Phi_e^{\mathtt{G}_y}(n) \downarrow$. Necessarily, $\Phi_e^{t_x}(n) \downarrow = \Phi_e^{t_y}(n) \downarrow = D(n)$. By extending q to agree with t on $[|q|, |t|)$ | if $|t| > |q$ we may assume that $|t| = |q = m$ and so $t \equiv_z q$. Let $l = |p|$. We now use both the interpolants guaranteed by Definition latrep 6.3.8(5) and the fact that p forces $\Phi_e^{\mathtt{G}_x}$ and $\Phi_e^{\mathtt{G}_y}$ to be total and equal.

For each k with $l \leq k < m$ we choose interpolants $\gamma_{k,i}$ (for $i \in \{1, 2, 3\}$) between $q(k)$ and $t(k)$ as in Definition latrep 6.3.8(5). We let $q_i(k) = p(k) = t(k)$ for $k < l$ and $q_i(k) = \gamma_{k,i}$ for $l \leq k < m$. We also let $q_0 = q$ and $q_4 = t$. So $q = q_0 \equiv_x q_1 \equiv_y q_2 \equiv_x q_3 \equiv_y q_4 = t$. We now extend the q_i in turn to make them force convergence at n but remain congruent modulo z. In fact, we make a single extension for all of them. By the fact that $p \Vdash (\Phi_e^{\mathtt{G}_x} = \Phi_e^{\mathtt{G}_y}$ and both are total) and $q_1 \leq p$, we can find an s_1 such that $q_1\hat{}s_1 \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = \Phi_e^{\mathtt{G}_y}(n) \downarrow$. We now, similarly, extend $q_2\hat{}s_1$ to $q_2\hat{}s_1\hat{}s_2$ such that $q_2\hat{}s_1\hat{}s_2 \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = \Phi_e^{\mathtt{G}_y}(n) \downarrow$. Finally we extend $q_3\hat{}s_1\hat{}s_2$ to $q_3\hat{}s_1\hat{}s_2\hat{}s_3 \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = \Phi_e^{\mathtt{G}_y}(n) \downarrow$. Let $s = s_1\hat{}s_2\hat{}s_3$ and consider $q_i\hat{}s$ for $i \leq 4$.

Looking at each successive pair we see, by the alternating (between x and y) congruences among the q_i, that they all force the same equal values for $\Phi_e^{\mathtt{G}_x}(n)$ and $\Phi_e^{\mathtt{G}_y}(n)$. (This follows from the transitivity of equality and preservation of either computations (use property from §codeTM 2.1) or forcing (Exercise ext 6.2.8) under extensions.)

Thus the first value, i, given by q and q_0 is equal to the last value, $D(n)$, given by t and q_4, as required.

By Theorem 6.2.6, there is a 2-generic $G \leq_T 0''$ (A''). As $G(\oplus A) \equiv_T G_1(\oplus A)$, we have that $f(1) \leq \mathbf{0}''$ $(\mathbf{a}'')$ as required. ■

We can now disprove the homogeneity conjecture for $\mathcal{D}' = \langle \mathcal{D}, \leq_T, ' \rangle$. This conjecture, like the analogous one for $\mathcal{D}$, was based on the empirical fact that every theorem about the degrees or the degrees with the jump operator relativizes and so if true in $\mathcal{D}$ (or $\mathcal{D}'$) then it is true in $\mathcal{D}(\geq \mathbf{c})$ or $\mathcal{D}'(\geq \mathbf{c})$ for every $\mathbf{c}$. The conjectures asserted then that $\mathcal{D} \cong \mathcal{D}(\geq \mathbf{c})$ and even that $\mathcal{D}' \cong \mathcal{D}'(\geq \mathbf{c})$ for every degree $\mathbf{c}$.

jhomc **Theorem 6.3.14** *There is* $\mathbf{c}$ *such that* $[\mathbf{0}, \mathbf{0}''] \ncong [\mathbf{c}, \mathbf{c}'']$ *and so* $(\mathcal{D}, \leq, ') \ncong (\mathcal{D}(\geq \mathbf{c}), \leq, ')$, *i.e. The homogeneity conjecture for* $\mathcal{D}'$ *fails.*

Proof. If not, then $[\mathbf{0}, \mathbf{0}''] \cong [\mathbf{c}, \mathbf{c}'']$ for every $\mathbf{c}$. To find a contradiction, it is sufficient (by Theorem 6.3.7) to find a partial lattice recursive in $\mathbf{c}$ which cannot be embedded in $[\mathbf{0}, \mathbf{0}'']$.

Now it is a fact of lattice theory that there are continuum many finitely generated lattices, indeed ones with only four generators. We supply such lattices with seven generators in the next section. On the other hand, only countably many finitely generated lattices can be embedded (as lattices) in $[\mathbf{0}, \mathbf{0}'']$ since the lattice embedded is determined by the image of its generators. Thus we may choose a lattice $\mathcal{L}$ which is finitely generated but not embeddable in $[\mathbf{0}, \mathbf{0}'']$. $\mathcal{L}$ has some degree, say $\mathbf{c}$. By Theorem 6.3.7, $\mathcal{L}$ is embeddable in $[\mathbf{c}, \mathbf{c}'']$. Thus $[\mathbf{0}, \mathbf{0}''] \ncong [\mathbf{c}, \mathbf{c}'']$ as required. ■

We will produce specific such degrees $\mathbf{c}$ in the next section and more examples will be provided in later Chapters.

Our usual question now is can we improve the complexity bound on the (top of the) embedding in Theorem 6.3.7. In particular, we might want the $f(1) \leq \mathbf{a}'$ or even $f(1)' = \mathbf{a}'$. Such improvements would also improve Theorem 6.3.14 by replacing $\mathbf{c}''$ by $\mathbf{c}'$. As was often the case in the constructions of Chapter 5, we can do this in two ways. The first asks for new dense sets (corresponding to new requirements) to add to those for 1-genericity that force us to preserve meets but can be meet recursively in $0'$. The second method says we should show, by some more clever argument, that the dense sets (requirements) for 1-genericity actually suffice to preserve meets as well.

For the first method, note that making the dense sets that force the preservation of meets to be uniformly recursive in $0'$ suffices get a generic G with $(\mathcal{P} \oplus G)' \leq_T G \oplus \mathcal{P}'$. (Here we use Propositions 6.1.25 and 6.2.17.)

latemb0' **Theorem 6.3.15** *If* $\mathcal{L}$ *is a partial lattice with* 0 *and* 1 *recursive in* A, *then there is an embedding* $f : \mathcal{L} \rightarrow \mathcal{D}[\mathbf{a}, \mathbf{a}']$ *with* $f(0) = \mathbf{a}$ *and* $f(1)' = \mathbf{a}'$. *Moreover, for* $x \in \mathcal{L}$, $f(x)$ *is uniformly recursive in* $f(1)$.

Proof. Again we consider the case that $A \equiv_T 0$ and so we have a recursive lattice table Θ for $\mathcal{L}$. We use the same notion of forcing and the same definition of our embedding as in Theorem parlatemb 6.3.7. We consider any 1-generic G which meets (in addition to the $\mathcal{C}_n$ of Proposition ngenericdense 6.2.10 that guarantee 1-genericity) some new dense sets $\mathcal{D}_e$. Now all the conditions, other than the preservation of meets, required to make f an embedding hold by the proof of Theorem parlatemb 6.3.7 and the 1-genericity of G. By our remarks before the statement of the Theorem, it suffices to show that the $\mathcal{D}_e$ are uniformly recursive in $0'$ and guarantee the preservation of meets.

For $e \in \mathbb{N}$ and $x \wedge y = z$ in $\mathcal{L}$, we define the new dense sets as follows:

$$\begin{aligned}
\mathcal{D}_{e,x,y} &= \{p | p \Vdash \exists n(\Phi_e^{\mathtt{G}_x}(n) \downarrow \neq \Phi_e^{\mathtt{G}_x}(n)) \text{ or} \\
(\exists n &< |p|)(\forall q \leq p)(q \nVdash \Phi_e^{\mathtt{G}_x}(n) \downarrow) \text{ or } (\exists n < |p|)(\forall q \leq p)(q \nVdash \Phi_e^{\mathtt{G}_y}(n) \downarrow) \text{ or} \\
(\forall r, s &\leq p)(\forall n)(\forall i, j)(s \equiv_z r \;\&\; s \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = i \;\&\; r \Vdash \Phi_e^{\mathtt{G}_y}(n) \downarrow = j \;\rightarrow i = j\}.
\end{aligned}$$

As the first clause in the definition of $\mathcal{D}_{e,x,y}$ is Σ_1 and the others are Π_1, these sets are clearly uniformly recursive in $0'$. The argument that meeting the $\mathcal{D}_{e,x,y}$ guarantees the preservation of meets is much as before but simpler. Suppose $\Phi_e^{G_x} = \Phi_e^{G_x}$ and both are total. Thus there is no $p \in \mathcal{G}$ satisfying any of the first three clauses in the definition of $\mathcal{D}_{e,x,y}$. So we may suppose that there is one satisfying the fourth clause. We now claim that we can compute $\Phi_e^{G_x} = \Phi_e^{G_x}$ recursively in G_z: For any n find a $q \leq p$ such that $q \equiv_z G$ and $q \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = i$. (There is one as before.) We claim that $\Phi_e^{G_y}(n) = i$. Again the point is that there is an $r \in \mathcal{G}$ and an j such that $r \Vdash \Phi_e^{\mathtt{G}_y}(n) \downarrow = j$ and, of course, $j = \Phi_e^{G_y}(n)$. We may, as above, assume that $|q| = |r|$ by lengthening one if necessary without changing the output of the forced computations. The fourth clause of the definition of $\mathcal{D}_{e,x,y}$ now guarantees that $i = j$ as required.

All that remains is to prove that the $\mathcal{D}_{e,x,y}$ are dense. Again, we can extract the proof from that of the previous Theorem. Given any p ask if there is a $q \leq p$ satisfying the fourth clause of the definition of $\mathcal{D}_{e,x,y}$. If so, we are done. If not, then for every $q \leq p$ there are $r, s \leq q$, n, i and j witnessing its failure, i.e. $s \equiv_z r \;\&\; s \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = i \;\&\; r \Vdash \Phi_e^{\mathtt{G}_y}(n) \downarrow = j$ but $i \neq j$. (Note that, by our conventions on congruences and forcing convergence, $n < |r| = |s|$.) Now follow the construction above to interpolate $\gamma_{k,i}$ (for $i \in \{1, 2, 3\}$) between $r(k)$ and $s(k)$ and get q_i for $i \leq 4$ starting with r, ending with s, all agreeing on $|p|$ and congruent modulo z. Attempt the extension procedure used above to get s_1, s_2, s_3. If at some point we are unable to find the next s_k to force convergence at n, then we have a condition extending p such that no extension forces convergence at n, i.e. an extension of p satisfying the second or third clause in the definition of $\mathcal{D}_{e,x,y}$. If we can find all the desired extensions, we argue much as before that one of the $q_i\hat{\ }s_1\hat{\ }s_2\hat{\ }s_3$ satisfies the first clause of the definition of $\mathcal{D}_{e,x,y}$ as required. ■

We now give the proof which shows that 1-genericity actually suffices to guarantee the embedding preserves meets.

emb<1gen **Theorem 6.3.16** *Given a recursive (in A) partial lattice $\mathcal{L}$ and any G 1-generic (over A) for the notion of forcing $\mathcal{P}$ of the proof of Theorem* parlatemb *6.3.7 the map from $\mathcal{L}$ to the degrees below that of $G(\oplus A)$ given by $x \longmapsto \deg(G_x)(\vee\mathbf{a})$ is a partial lattice embedding. More specifically, $G(\oplus A) \equiv_T G_1(\oplus A)$ whose degree is the top of the embedding is low (over A), i.e. $f(1)' = \mathbf{a}'$.*

Proof. Again we assume for convenience that $A \equiv_T 0$. We use the same forcing as in Theorem parlatemb 6.3.7, assume G is 1-generic and only have to show that meets are preserved.

Suppose that $x \wedge y = z$ and $\Phi_e^{G_x} = \Phi_e^{G_y} = D$. Consider the Σ_1 sets

$$T_e = \{t | t \Vdash \exists n(\Phi_e^{G_x}(n) \downarrow \neq \Phi_e^{G_y}(n) \downarrow)\}$$

and

$$\begin{aligned} S_{e,t} &= \{s : s \perp t \text{ or } (s \leq t \;\&\; \exists n, i, j \exists q, s_0, s_2, r \leq t(q \equiv_x s_0 \equiv_y s \equiv_x s_2 \equiv_y r \\ \&\; q &\Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = i \;\&\; r \Vdash \Phi_e^{\mathtt{G}_y}(n) \downarrow = j \;\&\; i \neq j))\}. \end{aligned}$$

By our assumptions there is no $t \subset G$ with $t \in T_e$ so let $t \subset G$ be such that $(\forall p \leq t)(p \notin T_e)$ and consider $S_{e,t}$. We first claim that no $s \subset G$ can be in $S_{e,t}$. Of course, no $s \subset G$ can be incompatible with $t \subset G$. So the only way we can have $s \in S_{e,t}$ is if $s \leq t$ and we have n, i, j, q, s_0, s_2 and r as in the definition of $S_{e,t}$. By our assumption that $\Phi_e^{G_x} = \Phi_e^{G_y} = D$, we may extend s to $\hat{s} \subset G$ such that $\hat{s} \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = \Phi_e^{\mathtt{G}_y}(n) \downarrow = D(n)$. we may then extend q, s_0, s_2 and r by the string ρ such that $s\hat{\ }\rho = \hat{s}$ to get $\hat{q}, \hat{s}_0, \hat{s}_2$ and $\hat{r}$ which also witness that $\hat{s} \in S_{e,t}$. By the definition of $\hat{s} \in S_{e,t}$, either $i \neq D(n)$ or $j \neq D(n)$. Suppose $i \neq D(n)$. As $q \equiv_x s_0$, $s_0 \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = i$ but as $s_0 \equiv_y s$, $s_0 \Vdash \Phi_e^{\mathtt{G}_y}(n) \downarrow = D(n)$ and so $s_0 \in T_e$ while $s_0 \leq t$ for a contradiction. The argument for $j \neq D(n)$ is similar.

Thus there is an $\hat{r} \subset G$ with no extension in $S_{e,t}$. In particular every extension of $\hat{r}$ is compatible with t and so $\hat{r} \supseteq t$, i.e. $\hat{r} \leq t$. We now claim that for any $q \leq \hat{r}$ with $q \equiv_z G$ such that $q \Vdash \Phi_e^{\mathtt{G}_x}(n) \downarrow = \Phi_e^{\mathtt{G}_y}(n) \downarrow = i$, $i = D(n)$ and so $D \leq_T G_z$ as required. If not consider a counterexample q. Let v be such that $r{=}\hat{r}\hat{\ }v = q$ so $q \equiv_z r$ and both extend $\hat{r}$. Choose interpolants $s_{0,}, s, s_2$ between q and r, i.e. $q \equiv_x s_0 \equiv_y s \equiv_x s_2 \equiv_y r$ with all extending $\hat{r}$. These conditions now provide witnesses that $s \in S_{e,t}$ for the desired contradiction as $s \leq \hat{r}$. ■

jhomc0' **Corollary 6.3.17** *There is $\mathbf{c}$ such that $[\mathbf{0}, \mathbf{0}'] \not\cong [\mathbf{c}, \mathbf{c}']$.*

latemb0'pr1 **Exercise 6.3.18** *If $\mathcal{L}$ is a recursive lattice with 0 and 1 then it can be embedded into $\mathcal{D}(\leq \mathbf{g})$ preserving both 0 and 1 for any Cohen 1-generic $\mathbf{g}$. (Recall Exercise* cohenvar *6.2.12.)*

We close this section with noting that not all constructions that can be done by adding on dense sets recursive in $0'$ to the ones for 1-genericity can be replaced by showing that 1genericity for the original forcing sufficed and investigating some variants of n-genericity suggested by the argument for Theorem emb<1gen 6.3.16.

Exercise 6.3.19 *If $H \leq_T 0'$ is 1-generic, then there is a uniformly recursive in $0'$ sequence $\mathcal{D}_e$ of dense sets (also for Cohen forcing) such that H does not meet all the $\mathcal{D}_e$. Indeed, if G meets all the $\mathcal{D}_e$ then $H \nleq_T G$.*

Definition 6.3.20 *Given a notion of forcing $\mathcal{P}$ and an $n \geq 1$, a filter $\mathcal{G}$ or descending sequence in $\mathcal{P}$ is weakly n-generic if it meets every dense $\Sigma_n^{\mathcal{P}}$ subset D of $\mathcal{P}$. The associated generic G is then also $\mathcal{P}$ weakly n-generic. A degree $\mathbf{g}$ is $\mathcal{P}$ weakly n-generic if it contains a $\mathcal{P}$ is weakly n-generic G.*

In the following Exercises, if no other notion of forcing is mentioned, Cohen forcing is intended.

Exercise 6.3.21 *For any $\mathcal{P}$, every weakly $(n+1)$-generic G or $\mathbf{g}$ is n-generic and every n-generic G or $\mathbf{g}$ is weakly $(n-1)$-generic. (For convenience we take weakly 0-generic to mean that it meets every recursive dense set)*

Exercise 6.3.22 *For any $\mathcal{P}$ and weakly $(n+1)$-generic G, if $X \leq_T \mathcal{P}^{(n)}, G$ then $X \equiv_T 0$.*

Exercise 6.3.23 *There is a weakly $(n+1)$-generic $\mathbf{g}$ which is not $(n+1)$-generic.*

Exercise 6.3.24 *There is an $(n+1)$-generic $\mathbf{g}$ which is not weakly $(n+2)$-generic.*

Thus the classes of weakly n-generic and n-generic degrees $\mathbf{g}$ form a strictly descending hierarchy of degree classes as n increases (with the n-generic degrees properly containing the weakly $(n+1)$-generic degrees).

Notes: Representations by equivalence relations is an old subject in lattice theory. In degree theory they were first used to embed all finite lattices in $\mathcal{D}$ and certain special lattices as initial segments of $\mathcal{D}$ by Thomason [1970]. The version used here in terms of tables is particularly suited to degree theory and was introduced in Lerman [1971] and extensively presented in his [1983]. Their use to embed lattices not as initial segments appears in Shore [1982] where it is used to prove Theorems 6.3.1, 6.3.7, 6.3.14, Corollary 6.3.17 and 6.3.15 as well as Exercise 6.3.15 and various strengthenings of Theorem 6.3.14 such as Corollary 6.3.17. The first proof of Theorem 6.3.14 and so the failure of the homogeneity conjecture for $\mathcal{D}'$ is due to Feiner [1970] but it depended on the construction of Σ_1 but not recursively presented Boolean algebras and known, but much more complicated, embeddings of lattices as initial segments of $\mathcal{D}$. Theorem 6.3.16 and Exercise 6.3.18 come from Greenberg and Montalbán [2003]. The notion of weakly n-generics and the Exercises about them are from Kurtz [1983]. ??Exercise 6.3.5 due to Paul Shafer (personal communication).??

ESS

6.4 Effective Successor Structures

For later applications, we would like to have a specific family of size $2^{\aleph_0}$ of finitely generated partial lattices that code arbitrary sets S in a relatively simple way and can be embedded below various degrees related to S in ways that we specify later. By Corollary extparlat 6.3.3, this will also supply the $2^{\aleph_0}$ many finitely generated lattices required for the proof of the failure of the homogeneity conjecture for the degrees with the jump operator (Theorem jhomc 6.3.14).

We begin our description of the desired partial lattices with ones that are *effective successor structures*.

ess **Definition 6.4.1** *An* effective successor structure (ess) *is a partial lattice with constant symbols e_0, e_1, d_0, f_0, f_1 generated by the corresponding elements and containing pairwise incomparable elements d_n such that, for each $n \geq 0$,*

$$(d_{2n} \vee e_0) \wedge f_1 = d_{2n+1} \qquad \text{and} \qquad (d_{2n+1} \vee e_1) \wedge f_0 = d_{2n+2}.$$

codeSrp **Definition 6.4.2** *If $\mathcal{L}$ is an ess and $g_0, g_1 \in \hat{\mathcal{L}}$ a partial lattice extension of $\mathcal{L}$, we say that* g_0, g_1 roughly code the set $S = \{n | d_n \leq g_0, g_1\}$ *in $\hat{\mathcal{L}}$. We say that* g_0, g_1 precisely code the set S *if, in addition, g_0, g_1 form an exact pair for the ideal generated by $\{d_n | n \in S\}$, i.e. every $x \leq g_0, g_1$ is below a finite join of some of the d_n.*

It is clear that given a set S there is an ess $\mathcal{L}$ and an extension $\mathcal{L}_S$ of $\mathcal{L}$ generated (over $\mathcal{L}$) by by two additional elements g_0, g_1 that precisely code S. (If this is not clear now, it will be when we consider the subclass of effective successor structures given in Definition ness 6.4.3 and Remark dnindep 6.4.4.) Thus the class of structures $\mathcal{L}_S$ provides us with continuum many different finitely generated partial lattices (and so lattices as well). Moreover, we have represented an arbitrary set S by a finitely generated partial lattice $\mathcal{L}_S$ in a reasonably simple way. For later applications we now analyze the relations between the complexities of S and $\mathcal{L}_S$ and so of their embeddings in $\mathcal{D}$. To make these relations as simple as possible we want to impose some additional conditions on our partial lattices and relax our notion of coding a set S.

ness **Definition 6.4.3** *A* nice effective successor structure (ness) *is a partial lattice extension of an ess containing various new elements and constants naming them: a, naming its greatest element, b, naming its least element, as well as c_0, c_1 and $\hat{d}_n$ for each $n \in \omega$ such that $c_0 \not\geq c_1$ and $(\forall n \in \omega)(d_n \vee c_0 \geq c_1$ & $d_n \wedge \hat{d}_n = b$ & $(\forall m \neq n)(\hat{d}_n \geq d_m)$.*

dnindep **Remark 6.4.4** *In any ness $\mathcal{L} \subseteq \mathcal{D}$ the degrees $\mathbf{d}_n$ (denoted by the constants d_n) are independent: no finite join $\mathbf{x}$ of $\mathbf{d}_m$'s for $m \neq n$ can be above $\mathbf{d}_n$ (denoted by $\hat{d}_n$) as $\mathbf{x} \leq \hat{\mathbf{d}}_n$ while $\mathbf{d}_n \wedge \hat{\mathbf{d}}_n = \mathbf{b}$ (denoted by b) implies that $\mathbf{d}_n \not\leq \mathbf{d}_n$. Thus, for any set S, $\mathcal{L}$ can be extended to an $\mathcal{L}_S \subseteq \mathcal{D}$ by adding on an exact pair $\mathbf{g}_0, \mathbf{g}_1$ for the ideal generated by the $\mathbf{d}_n$ for $n \in S$ so that S is precisely coded by $\mathbf{g}_0, \mathbf{g}_1$ in $\mathcal{L}_S$.*

We now want to analyze the complexity of sets coded in a related manner in such substructures of $\mathcal{D}$. As $\wedge$ on degrees is more complicated than $\leq$ or $\vee$, our first goal is to eliminate the use of $\wedge$ (in defining the d_n from the generators) in favor of an approximation in terms of just $\leq$ and $\vee$ which will be sufficient to code a set S. Moreover, as the use of $\not\leq$ also increases the complexity of formulas in $\mathcal{D}$, we do not want to use it either. We provide such an approximation $\varphi_n(x)$ to d_n in an arbitrary ness.

Definition 6.4.5 *A formula θ of a language $\mathcal{F}$ is positive Σ_1 if it is built from the atomic formulas of $\mathcal{F}$ using only conjunction, disjunction and existential quantification.*

approxdn **Proposition 6.4.6** *If $\mathcal{L}$ is a ness then there is a recursive sequence $\varphi_n(x)$ of positive Σ_1 formulas in the language containing only $\vee$, $\leq$ and the constants $c_0, c_1, d_0, e_0, e_1, f_0, f_1$ such that the following two assertions are true in $\mathcal{L}$ for every n.*

1. *$\varphi_n(d_n)$.*
2. *$\forall x(\varphi_n(x) \rightarrow b < x \leq d_n)$.*

Moreover, if $\hat{\mathcal{L}}$ is a partial lattice extension of $\mathcal{L}$, then the same facts are true in $\hat{\mathcal{L}}$.

Proof. We define φ_n by recursion on n. We begin with $\varphi_0(x) \equiv x = d_0$. The successor steps depend on their parity.

$$\begin{aligned} \varphi_{2n+1}(x) &\equiv x \vee c_0 \geq c_1 \ \&\ \exists y(\varphi_{2n}(y) \ \&\ x \leq (y \vee e_0), f_1). \\ \varphi_{2n+2}(x) &\equiv x \vee c_0 \geq c_1 \ \&\ \exists y(\varphi_{2n+1}(y) \ \&\ x \leq (y \vee e_1), f_0). \end{aligned}$$

It is clear by induction that the φ_n are positive Σ_1 formulas with only the desired constants. We prove the two assertions of the Proposition by induction. Both are obvious for φ_0. Consider φ_{2n+1} and the first assertion. By Definition ness 6.4.3, $d_{2n+1} \vee c_0 \geq c_1$. We claim that d_{2n} is the desired witness y that the second clause of $\varphi_{2n+1}(d_{2n+1})$ holds as well: by induction $\varphi_{2n}(d_{2n})$ and, by Definition ness 6.4.3, $d_{2n+1} \leq (d_{2n} \vee e_0), f_1$ as required.

Now, for the second assertion, suppose that $\varphi_{2n+1}(x)$ holds. As $x \vee c_0 \geq c_1$ and $c_0 \not\geq c_1$, $b < x$. Let y be the witness that the second clause of $\varphi_{2n+1}(x)$ holds. Thus $\varphi_{2n}(y)$ and so by induction $y \leq d_{2n}$. Finally, by the second clause in φ_{2n+1}, $x \leq (d_{2n} \vee e_0) \wedge f_1 = d_{2n+2}$ as required. The argument for φ_{2n+2} is essentially the same.

The same arguments work in any $\hat{\mathcal{L}}$ extending $\mathcal{L}$. ■

For later use in complexity calculations (starting with Proposition codeSs3 6.4.9), we now point out that we can replace all the existential quantifiers in the φ_n by bounded ones.

qtbdd **Proposition 6.4.7** *With the notation of Proposition approxdn 6.4.6, if we replace all the existential quantifiers $\exists y$ in φ_n (for any n) by bounded quantifiers $\exists y \leq a$ to get formulas $\hat{\varphi}_n$, then $\varphi_n(x)$ and $\hat{\varphi}_n(x)$ are equivalent in every $\hat{\mathcal{L}}$ extending $\mathcal{L}$.*

Proof. Fix any $\hat{\mathcal{L}}$ extending $\mathcal{L}$ and proceed by induction on n. The first formula φ_1 has no quantifiers so there is nothing to prove. For φ_{2n+1} as we may assume that we have replaced all the quantifiers in φ_{2n} by bounded ones, the only quantifier that we need to consider is $\exists y$ in the second clause $\exists y(\varphi_{2n}(y)\ \&\ x \le (y \vee e_0), f_1)$. By Proposition 6.4.6, $\varphi_{2n}(y) \rightarrow y \le d_{2n}$ and so this clause is equivalent to $(\exists y \le a)(\varphi_{2n}(y)\ \&\ x \le (y \vee e_0), f_1)$. The argument for φ_{2n+2} just replaces $2n$ by $2n+1$. ■

We now adjust our notion of coding a set S by using the φ_n in place of the d_n and then calculate the complexity of sets coded in $\mathcal{D}$.

codeS **Definition 6.4.8** *If $\mathcal{L}$ is a ness and $g_0, g_1 \in \hat{\mathcal{L}}$ a partial lattice extension of $\mathcal{L}$, we say that g_0, g_1* code the set *$S = \{n | \hat{\mathcal{L}} \vDash \exists x(\varphi_n(x)\ \&\ x \le g_0, g_1)\}$ in $\hat{\mathcal{L}}$.*

codeSs3 **Proposition 6.4.9** *Given a ness $\mathcal{L} \subseteq \mathcal{D}$ and any degrees $\mathbf{g}_0, \mathbf{g}_1$, the set S coded by $\mathbf{g}_0, \mathbf{g}_1$in $\mathcal{D}$ is Σ_3 in $A \oplus G_0 \oplus G_1 = Z \in \mathbf{z}$ and is the same set coded by $\mathbf{g}_0, \mathbf{g}_1$ in any extension $\hat{\mathcal{L}}$ of $\mathcal{L}$ in $\mathcal{D}$ containing $\{\mathbf{x} | \mathbf{x} \le \mathbf{z}\}$. (As might be expected we are taking the degrees denoted by the constants of the ness to have the natural names: a denotes $\mathbf{a}$, b denotes $\mathbf{b}$, etc. We also follow the usual convention of naming representatives of the degrees: $A \in \mathbf{a}$, $B \in \mathbf{b}$, etc.)*

Proof. By Proposition 6.4.7, $\varphi_n(x)$ and its bounded version $\hat{\varphi}_n(x)$ are equivalent in any $\hat{\mathcal{L}}$ as in our Proposition. Thus we may replace the φ_n by these $\hat{\varphi}_n$ in the definition of the set S coded by $\mathbf{g}_0, \mathbf{g}_1$ in $\hat{\mathcal{L}}$. As by Proposition 6.4.6, they can hold only of $x \le \mathbf{a}$ and all the constants in these formulas are below $\mathbf{z}$ and the quantifiers are bounded by $\mathbf{a} \le \mathbf{z}$, it is clear that the truth of the defining formula $\exists x(\varphi_n(x)\ \&\ x \le g_0, g_1)$ for $n \in S$ is independent of our choice of $\hat{\mathcal{L}}$ as long as it contains $\{\mathbf{x} | \mathbf{x} \le \mathbf{z}\}$. So we consider $\hat{\mathcal{L}} = \{\mathbf{x} | \mathbf{x} \le \mathbf{z}\}$.

We can represent all the degrees below $\mathbf{z}$ by sets of the form Φ_i^Z and, in particular we choose indices a, b etc. so that $\Phi_a^Z \in \mathbf{a}$, $\Phi_b^Z \in \mathbf{b}$, etc. Conversely, every total Φ_i^Z has degree $\le \mathbf{z}$. We now translate the defining formulas $\theta_n = \exists x(\hat{\varphi}_n(x)\ \&\ x \le g_0, g_1)$ that code whether n is in S into the language of Turing reducibility on sets recursive in Z and so sentences $\hat{\theta}_n$ of arithmetic. First, recall the uniform procedures for dealing with join and changing oracles for sets recursive in Z from Examples 2.1.10 and 2.1.11: $\Phi_e^Z \oplus \Phi_i^Z = \Phi_{p(e,i)}^Z$ and $\Phi_e^{\Phi_i^Z} = \Phi_{t(e,i)}^Z$. We begin our translation of the θ_n by choosing, for each constant symbol v, a number $\hat{v}$ such that $\Phi_{\hat{v}}^Z \in \mathbf{v}$. We now replace each occurrence of v in θ_n by $\Phi_{\hat{v}}^Z$. We next inductively replace each bounded quantifier $(\exists y \le a)\psi(y)$ by $\exists i(\Phi_{t(i,\hat{a})}^Z$ is total & $\psi(\Phi_{t(i,\hat{a})}^Z))$. Then we inductively eliminate $\vee$ by replacing terms of the form $\Phi_e^Z \vee \Phi_i^Z$ by $\Phi_{p(e,i)}^Z$. Finally, we replace all atomic formulas of the form $\Phi_e^Z \le \Phi_i^Z$ or $\Phi_e^Z = \Phi_i^Z$ by $\Phi_e^Z \le_T \Phi_i^Z$ and $\Phi_e^Z \equiv_T \Phi_i^Z$, respectively. It is clear that the resulting sentence $\hat{\theta}_n$ (of arithmetic) is equivalent to the truth of θ_n in $\hat{\mathcal{L}}$. As $\Phi_{t(i,\hat{a})}^Z$ being total is a Π_2^Z property (Example 4.4.3) while $\Phi_e^Z \le_T \Phi_i^Z$ and $\Phi_e^Z \equiv_T \Phi_i^Z$ are Σ_s^Z relations (Example 4.4.4), $\hat{\theta}_n$ is Σ_3^Z as required. ■

setscoded **Remark 6.4.10** *For any degree $\mathbf{d}$ and ness $\mathcal{L}$ contained in $\mathcal{D}(\le \mathbf{d})$, if we can show that all Σ_3^D sets are coded in $\mathcal{L}$ by pairs below $\mathbf{d}$, then we know that these sets are exactly* *those which are Σ_3^D. Similarly, if we have a ness $\mathcal{L} \subseteq \hat{\mathcal{L}} \subseteq \mathcal{D}$ such that, for every $\mathbf{d} \in \hat{\mathcal{L}}$, every set Σ_3^D is coded in $\mathcal{L}$ by a pair in $\hat{\mathcal{L}}$, then the sets so coded are precisely the sets Σ_3^D for any $\mathbf{d} \in \hat{\mathcal{L}}$.*

These calculations will play a crucial role in our determination of the complexity of the theories of $\mathcal{D}$ and various substructures such as the degrees below $\mathbf{0}'$ as well as our global results on definability and automorphisms of $\mathcal{D}$ and its substructures in various ?? later chapters.

We give one application now that determines specific counterexamples to the homogeneity conjecture for $\mathcal{D}'$. Stronger results will be provided in ??.

jhomcc **Corollary 6.4.11** *The degrees below $\mathbf{0}'$, $[\mathbf{0}, \mathbf{0}']$, are not isomorphic to $[\mathbf{0}^{(5)}, \mathbf{0}^{(6)}]$ and so $\mathcal{D}(\leq,') \not\cong \mathcal{D}(\geq \mathbf{0}^{(5)})(\leq,')$.*

Proof. By Theorem latemb0' 6.3.15 or emb<1gen 6.3.16 (relativized to $0^{(5)}$), there is a ness $\mathcal{L}$ and degrees $\mathbf{g}_0, \mathbf{g}_1$ in $[\mathbf{0}^{(5)}, \mathbf{0}^{(6)}]$ which codes $0^{(5)}$. However, by Proposition codeSs3 6.4.9, any such substructure in $[\mathbf{0}, \mathbf{0}']$ codes a set which is $\Sigma_3^{0'}$ and so Σ_4. Of course, $0^{(5)} \notin \Sigma_4$. ■

Exercise 6.4.12 *Improve Corollary*jhomcc *6.4.11 by replacing $[\mathbf{0}^{(5)}, \mathbf{0}^{(6)}]$ by $[\mathbf{0}^{(4)}, \mathbf{0}^{(5)}]$. Hint: Code both $0^{(4)}$ and its complement.*

Exercise 6.4.13 *Show that $\mathcal{D}(\leq,') \not\cong \mathcal{D}(\geq \mathbf{0}^{(3)})(\leq,')$. Hint: Use the fact that Theorem* letemb1gen **??** *embeds lattices recursive in A so that the top of the lattice has jump A'.*

We conclude by noting that we can always restrict ourselves to coding by exact pairs in substructures of $\mathcal{D}$ and so to precise coding.

codeexactp **Proposition 6.4.14** *If $\mathcal{L} \subseteq \mathcal{D}$ is a ness, $S \subseteq \mathbb{N}$ and $\mathbf{g}_0$ and $\mathbf{g}_1$ are an exact pair for the ideal generated by the $\mathbf{d}_n$ for $n \in S$, i.e. $\mathbf{g}_0$, $\mathbf{g}_1$ precisely code S, then S is the set coded by $\mathbf{g}_0$ and $\mathbf{g}_1$ in $\mathcal{D}$ (and so in any $\hat{\mathcal{L}} \subseteq \mathcal{D}$ containing the degrees below $\mathbf{a} \vee \mathbf{g}_0 \vee \mathbf{g}_1$).*

Proof. Let $\hat{S}$ be the set coded by $\mathbf{g}_0$, $\mathbf{g}_1$ in $\mathcal{D}$. Clearly if $n \in S$, $\mathbf{d}_n \leq \mathbf{g}_0, \mathbf{g}_1$. By Proposition approxdn 6.4.6, $\mathcal{D} \vDash \varphi_n(\mathbf{d}_n)$ and so by Definition codeS 6.4.8, $n \in \hat{S}$. On the other hand , if $n \in \hat{S}$, then, again by Definition codeS 6.4.8, there is an $\mathbf{x} \leq \mathbf{g}_0, \mathbf{g}_1$ such that $\mathcal{D} \vDash \varphi_n(\mathbf{x})$. The first of these facts tells us that $\mathbf{x}$ is below the join of finitely many $\mathbf{d}_m$ for $m \in S$. The second that $\mathbf{x} \leq \mathbf{d}_n$. As the $\mathbf{d}_i$ are independent by Remark dnindep 6.4.4, $n \in S$ as required. That the same set S is coded in any $\hat{\mathcal{L}} \subseteq \mathcal{D}$ containing the degrees below $\mathbf{a} \vee \mathbf{g}_0 \vee \mathbf{g}_1$ follows from Proposition codeSs3 6.4.9. ■

Notes: The conditions on (nice) effective successor structures and their use in coding arithmetic come from Shore [1981] as does Proposition codeSs3 6.4.9. ??other references for some of the development of these ideas??

Chapter 7

The Theories of $\mathcal{D}$ and $\mathcal{D}(\leq \mathbf{0}')$

In the previous chapter, we talked about embeddability issues. We need to consider more in order to understand the theory of the degrees. We now approach theorems which say that the theories of (i.e. the sets of sentences true in) $\mathcal{D}$ and $\mathcal{D}(\leq 0')$ are as complicated as possible. More precisely they are of the same Turing (even $1-1$) degree as true second and first order arithmetic, respectively. The method used is interpreting arithmetic in the degree structures.

interp

7.1 Interpreting Arithmetic

We say that we can interpret (true first order) arithmetic in a structure $\mathcal{S}$ with parameters $\bar{p}$ if there are formulas $\varphi_D(x)$, $\varphi_+(x,y,z)$, $\varphi_\times(x,y,z)$, $\varphi_<(x,y)$ all with parameters $\bar{p}$ and one $\varphi_c(\bar{p})$ such that for any $\bar{p} \in \mathcal{S}$ such that $\mathcal{S} \vDash \varphi_c(\bar{p})$ the structure $\mathcal{M}(\bar{p})$ with domain $D(\bar{p}) = \{x \in \mathcal{S} | \mathcal{S} \vDash \varphi_D(x)\}$ and relations $+$, $\times$ and $<$ defined by $\varphi_+(x,y,z)$, $\varphi_\times(x,y,z)$, $\varphi_<(x,y)$, respectively, is isomorphic to true arithmetic, i.e. the natural numbers $\mathbb{N}$ with relations given by $+$, $\times$ and $<$ respectively and there is at least one such $\bar{p}$. (We are writing the operations $+$ and $\times$ in relational form $+(x,y,z) \Leftrightarrow x+y=z$ and similarly for $\times$.) In this situation, the theory of true first order arithmetic, $Th(\mathbb{N})$, i.e. the set of sentences of arithmetic in this language true in $\mathbb{N}$, is reducible to $Th(\mathcal{S})$, the set of sentences in the language of $\mathcal{S}$ true in S. Indeed, the reduction is a $1-1$ reduction. More precisely there is a recursive function T taking sentences φ of arithmetic to ones φ^T of $\mathcal{S}$ such that $\mathbb{N} \vDash \varphi \Leftrightarrow \mathcal{S} \vDash \forall \bar{p}(\varphi_c(\bar{p}) \rightarrow \varphi^T)$. The definition of T is given by induction. Atomic formulas $+(x,y,z)$, $\times(x,y,z)$ and $x < y$ are taken to $\varphi_+(x,y,z)$, $\varphi_\times(x,y,z)$, $\varphi_<(x,y)$, respectively. A formula of the form $\exists w \psi$ is taken to $\exists w(\varphi_D(w) \;\&\; \psi^T)$ while $\forall w \psi$ is taken to $\forall w(\varphi_D(w) \;\rightarrow\; \psi^T)$. It should be clear (and, if not, routine to prove) by induction that if $\mathcal{M}(\bar{p}) \cong \mathbb{N}$ then, any sentence φ (of the relational formulation of arithmetic) is true in $\mathbb{N}$ if and only if φ^T is true in $\mathcal{M}(\bar{p})$. Thus if $\varphi_c(\bar{p})$ guarantees that $\mathcal{M}(\bar{p}) \cong \mathbb{N}$, we have the desired recursive reduction from $Th(\mathbb{N})$ to $Th(\mathcal{S})$.

A second order structure is a two sorted structure (i.e. one with two sorts of variables say x and X in its language and two domains U and $W \subseteq 2^U$ over which the two types of variable range, respectively. This provides the semantics for the quantifiers $\exists x, \forall x, \exists X$, and $\forall X$ in the obvious way). The language also has relation symbols and relations on the first sort as in a standard first order language and structure. In addition, it has one relation $x \in X$ between elements of the first sort and ones of the second sort that is interpreted by true membership. We say that it is a true second order structure if $W = 2^U$, i.e. the second order quantifiers range over all subsets of the domain U of the usual first order structure. It is a model of true second order of arithmetic if $U = \mathbb{N}$, the first order language is that of arithmetic as above and $W = 2^{\mathbb{N}}$. (Note that as with true first order arithmetic there is, up to isomorphism, only one model of true second order of arithmetic.)

We extend our notion of an interpretation of arithmetic to second order structures by adding a formula $\varphi_S(x, \bar{y})$ which implies $\varphi_D(x)$. For each tuple of degrees $\bar{\mathbf{y}}$, we are thinking of $\varphi_S(x, \bar{\mathbf{y}})$ as defining the set of $n \in \mathbb{N}$ such that $\varphi_S(\mathbf{d}_n, \bar{\mathbf{y}})$ holds for $\mathbf{d}_n$ the degree corresponding to the nth element of the model in the ordering given by $\varphi_<$, We then translate the second order quantifiers by replacing each atomic formula $x \in X$ by $\varphi_S(x, \bar{y}_X)$, $\exists X \psi$ by $\exists \bar{y}_X \psi^T$ and $\forall X \psi$ by $\forall \bar{y}_X \psi^T$ where we are thinking of the $\bar{y}_X$ as coding the set X. If, as before, $\varphi_c(\bar{p})$ guarantees that the associated first order structure is isomorphic to $\mathbb{N}$ and, in addition, as $\bar{y}$ ranges over S^n (where n is the length of $\bar{y}$) the sets $S_{\bar{y}} = \{x | \varphi_S(x, \bar{y})\}$ range exactly over all subsets of $D(\bar{p})$ then it clear (or routine to prove) that, for any second order sentence φ of arithmetic, φ is satisfied in the true second order model of arithmetic if and only if $\mathcal{S} \vDash \varphi_c(\bar{p}) \rightarrow \varphi^T$. In this case we again have a recursive reduction: a sentence ψ of second order arithmetic is "true", i.e. satisfied in the model of true second order of arithmetic if an only if $\mathcal{S} \vDash \forall \bar{p}(\varphi_c(\bar{p}) \rightarrow \psi^T)$.

Our goals now are to prove that there are interpretations of true second order arithmetic in $\mathcal{D}$ and true first order arithmetic in $\mathcal{D}(\leq \mathbf{0}')$. The first we complete in this chapter. We actually show in the next section that we can code and quantify over all countable relations on $\mathcal{D}$ in a first order way by quantifying over elements of $\mathcal{D}$. From this result is routine to get a coding as described here of second order arithmetic in $\mathcal{D}$. The results and analysis need for $\mathcal{D}(\leq \mathbf{0}')$ are mostly contained in this chapter but the proof also requires material from the next chapter as well. In each case, the correctness condition $\varphi_c(\bar{p})$ includes the translations (via T) of the axioms of a finite axiomatization of arithmetic such as Robinson arithmetic that is strong enough to guarantee that any model of the axioms in which the ordering $<$ on its domain is isomorphic to ω is actually isomorphic to $\mathbb{N}$. The crucial steps are then to prove that there are $\bar{p}$ such that $\mathcal{M}(\bar{p}) \cong \mathbb{N}$ and that there is a formula $\varphi_{\hat{c}}$ which guarantees that the ordering of $\mathcal{M}(\bar{p})$ (given by $\varphi_<(\bar{p})$) is isomorphic to ω.

We begin with $\mathcal{D}$ and coding countable subsets of pairwise incomparable degrees by using Slaman-Woodin forcing. We then show how to deal with arbitrary countable relations on degrees.

7.2 Slaman-Woodin Forcing and $Th(\mathcal{D})$

Let $\mathbf{S} = \{\mathbf{c}_i | i \in \mathbb{N}\}$ be a countable set of pairwise incomparable degrees. We want to make $\mathbf{S}$ definable in $\mathcal{D}$ from three parameters $\mathbf{c}$, $\mathbf{g}_0$ and $\mathbf{g}_1$. The definition is that $\mathbf{S}$ is the set of minimal degrees $\mathbf{x} \leq \mathbf{c}$ such that $(\mathbf{x} \vee \mathbf{g}_0) \wedge (\mathbf{x} \vee \mathbf{g}_1) \neq \mathbf{x}$ in the strong sense that there is a $\mathbf{d} \leq \mathbf{x} \vee \mathbf{g}_0, \mathbf{x} \vee \mathbf{g}_1$ such that $\mathbf{d} \not\leq \mathbf{x}$.

$\boxed{\texttt{SW}}$ **Theorem 7.2.1** *For any set $S = \{C_0, C_1, \ldots\}$ of pairwise Turing incomparable subsets of $\mathbb{N}$ let $C = \oplus C_i$. There are then G_0, G_1 and D_i such that, for every $i \in \mathbb{N}$ and $j < 2$, $D_i \leq_T C_i \oplus G_j$ while $D_i \not\leq_T C_i$. Moreover, the C_i are minimal with this property among sets recursive in C in the sense that for any $X \leq_T C$ for which there is a D such that $D \leq_T X \oplus G_j$ $(j < 2)$ but $D \not\leq_T X$ there is an i such that $C_i \leq_T X$. Indeed, there is a notion of forcing $\mathcal{P}$ recursive in C such that any 2-generic computes such G_0 and G_1. Thus for $\mathbf{c}_i, \mathbf{c}$ and $\mathbf{g}_0, \mathbf{g}_1$ the degrees of C_i, C, G_0 and G_1 respectively, the set $\mathbf{S} = \{\mathbf{c}_i | i \in \mathbb{N}\}$ is definable in $\mathcal{D}$ from the three parameters $\mathbf{c}$, $\mathbf{g}_0$ and $\mathbf{g}_1$.*

Proof. Without loss of generality we may assume that each C_i is recursive in any of its infinite subsets: simply replace C_i by the set of binary stings σ such that $\sigma \subset C_i$. The point of this assumption is that to compute C_i from some X it suffices to show that X can enumerate an infinite subset of C_i as then there is an infinite subset of this set recursive in X and so then is C_i.

We build G_i as required by forcing in such a way as to uniformly define the D_i from G_0 and C_i and such that D_i is also recursive in $G_1 \oplus C_i$ (although not uniformly). We begin with the coding scheme that says how we compute the D_i.

Let $\{c_{i,0}, c_{i,1}, \ldots\}$ list C_i in increasing order. Our plan is that $D_i(n)$ should be $G_0(c_{i,n})$ and so the D_i are uniformly recursive in $G_0 \oplus C_i$. We call $\langle i, k \rangle$ a *coding location for C_i* if $k \in C_i$. To make sure that $D_i \leq_T G_1 \oplus C_i$ as well, we guarantee that $G_0^{[i]}(c_n) = G_1^{[i]}(c_n)$ for all but finitely many n. We now turn to our notion of forcing $\mathcal{P}$.

The forcing conditions p are triples of the form $\langle p_0, p_1, F_p \rangle$ where $p_0, p_1 \in 2^{<\omega}$, $|p_0| = |p_1|$, and F_p is a finite subset of ω. We let the length of condition p be $|p| = |p_0| = |p_1|$. Refinement is defined by

$$\begin{aligned} p \leq q \ \Leftrightarrow \ & p_0 \supseteq q_0, p_1 \supseteq q_1, F_p \supseteq F_q, \text{ and} \\ & \text{if } i \in F_q \text{ and } |q| < \langle i, c_{i,n} \rangle \leq |p| \text{ then } p_0(\langle i, c_{i,n} \rangle) = p_1(\langle i, c_{i,n} \rangle). \end{aligned}$$

This is a finite notion of forcing with extension recursive in C. The function V is defined in the obvious way: $V(p) = p_0 \oplus p_1$ so our generic object defined from a filter $\mathcal{G}$ is $G_0 \oplus G_1$ where $G_k = \cup\{p_k | p \in \mathcal{G}\}$. We use $\mathtt{G}_k$ in our language to mean the k^{th} coordinate the generic object. Note that $C \leq_T \mathcal{P}$ as well (Exercise) and so n-generic for $\mathcal{P}$ means generic for all Σ_n^C sets.

Note that for any $\varphi \in \Sigma_1$, if $p \Vdash \varphi$ then $(p_0, p_1, \emptyset) \Vdash \varphi$ as $V(p) = V(\langle p_0, p_1, \emptyset \rangle$. So if $q \leq p$ and $q \Vdash \psi$ for $\psi \in \Sigma_1$ then $(q_0, q_1, F_P) \Vdash \psi$ as well.

Suppose that $\mathcal{G}$ is 1-generic for $\mathcal{P}$. It is immediate from the definition of $\leq_{\mathcal{P}}$ and the density of the recursive (in $\mathcal{P}$) sets $\{p|i \in F_p\}$ that $G_0^{[i]}$ and $G_1^{[i]}$ differ on at most finitely many $n \in C_i$. (If $i \in F_p$ and $p \in \mathcal{G}$ then $G_0^{[i]}(m) = G_1^{[i]}(m)$ for $m \in C_i$ and $m > |p|$.) Thus $D_i \leq_T G_1 \oplus C_i$ as required.

We next show that $D_i \not\leq_T C_i$, that is $\Phi_e^{C_i} \neq D_i$ for each e. Suppose for the sake of a contradiction that $D_i = \Phi_e^{C_i}$ for some e (and so in particular $\Phi_e^{C_i}$ is total). Consider the Σ_1^C set

$$S_{i,e} = \{p : \exists m(p_0(\langle i, c_{i,m}\rangle) \neq \Phi_e^{C_i}(m))\}.$$

The $S_{i,e}$ are dense because if $p \in P$ and m is such that $\langle i, c_{i,m}\rangle > |p|$ then we can define $q \leq p$ by $F_q = F_p$ and for $|p| \leq j \leq \langle i, c_{i,m}\rangle$ put $q_0(j) = q_1(j) = 1 - \Phi_e^{C_i}(m)$. So $q \in S_{i,e}$ and $q \leq p$ as desired. Thus, there is a $p \in \mathcal{G} \cap S_{i,e}$ for which

$$D_i(m) = G_0(\langle i, c_{i,m}\rangle) = p_0(\langle i, c_{i,m}\rangle) \neq \Phi_e^{C_i}(m),$$

contradicting $D_i = \Phi_e^{C_i}$.

Now, we have to ensure the minimality of the C_i. In other words, we want to prove that if

$$\Phi_e^{X\oplus G_0} = \Phi_i^{X\oplus G_1} = D, \qquad X \leq_T C \qquad \text{and} \qquad D \not\leq_T X$$

then $C_k \leq_T X$ for some k.

Consider the sentence φ that says that $\Phi_e^{X\oplus G_0}$ and $\Phi_i^{X\oplus G_1}$ are total and equal. It is Π_2 in C (because $X \leq_T C$) and true of $G = G_0 \oplus G_1$. So, if we now assume that $\mathcal{G}$ is 2-generic, there is $p \in \mathcal{G}$ such that $p \Vdash \varphi$. Suppose first that $\neg\exists n(\exists\sigma \supseteq p_0)(\exists\tau \supseteq p_0)[\Phi_e^{X\oplus\sigma}(n) \downarrow\neq \Phi_e^{X\oplus\tau}(n)\downarrow]$. Then we claim D is computable from X. To compute $D(n)$ search for any $\sigma \supseteq p_0$ such that $\Phi_e^{X\oplus\sigma}(n)\downarrow$ and output this value as the answer. There is such a $\sigma \subset G_0$ by the totality of $\Phi_e^{X\oplus G_0}$. Our assumption that there is no pair of extensions of p_0 that give two different answers implies that any such σ gives the answer $\Phi_e^{X\oplus G_0}(n) = D(n)$.

On the other hand, suppose there is such a splitting for n given by $p_0\hat{}\sigma$, $p_0\hat{}\tau$. By extending one of σ and τ if necessary, we may assume that $|\sigma| = |\tau|$. We claim that $p_0\hat{}\sigma$ and $p_0\hat{}\tau$ differ at a coding location $\langle k, c_{k,m}\rangle$ for some $k \in F_p$. Let τ' be such that

$$\Phi_i^{X\oplus(p_1\hat{}\tau\hat{}\tau')}(n)\downarrow = \Phi_e^{X\oplus(p_0\hat{}\tau\hat{}\tau')}(n)\downarrow.$$

There must be such a τ' as $(p_0\hat{}\tau, p_1\hat{}\tau, F_p) \leq p$ and so it has a further extension $q = (p_0\hat{}\tau\hat{}\rho_0, p_1\hat{}\tau\hat{}\rho_1, F_p)$ which forces $\Phi_e^{X\oplus G_0}(n)\downarrow = \Phi_i^{X\oplus G_1}(n)\downarrow$. Next consider $\hat{q} = (p_0\hat{}\tau\hat{}\rho_0, p_1\hat{}\tau\hat{}\rho_0, F_p) \leq p$. It also has an extension $(p_0\hat{}\tau\hat{}\rho_0\hat{}\mu_0, p_1\hat{}\tau\hat{}\rho_0\hat{}\mu_1, F_p) \Vdash \Phi_e^{X\oplus G_0}(n)\downarrow = \Phi_i^{X\oplus G_1}(n)\downarrow$. It is now clear that $\tau' = \rho_0\hat{}\mu_1$ has the desired property. Next, consider the condition $q = (p_0\hat{}\sigma\hat{}\tau', p_1\hat{}\tau\hat{}\tau', F_p)$. Notice that $q \not\leq p$ because:

1. $\Phi_e^{X\oplus(p_0\hat{}\sigma)}(n) = \Phi_e^{X\oplus(p_0\hat{}\sigma\hat{}\tau')}(n)$ as $p_0\hat{}\sigma\hat{}\tau' \supseteq p_0\hat{}\sigma$.

2. $\Phi_i^{X\oplus(p_1\hat{}\tau\hat{}\tau')}(n) = \Phi_e^{X\oplus(p_0\hat{}\tau)}(n)$ by our choice of τ', but

3. $\Phi_e^{X\oplus(p_0\hat{}\sigma)}(n) \neq \Phi_e^{X\oplus(p_0\hat{}\tau)}(n)$ because $n, p_0\hat{}\sigma, p_0\hat{}\tau$ were chosen to be splitting.

Hence, $\Phi_e^{X\oplus(p_0\hat{}\sigma\hat{}\tau')}(n) \neq \Phi_i^{X\oplus(p_1\hat{}\tau\hat{}\tau')}(n)$ and so q does not extend p. However, $p_0\hat{}\sigma\hat{}\tau' \supseteq p_0$ and $p_1\hat{}\tau\hat{}\tau' \supseteq p_1$, so it must be that $p_0\hat{}\sigma\hat{}\tau'$ and $p_1\hat{}\tau\ \hat{}\tau'$ differ at a coding location above $|p|$. Therefore, $p_0\hat{}\sigma$ and $p_0\hat{}\tau$ differ at a coding location $\langle k, n\rangle$ with $k \in F_p$.

We now show that there must be such $p_0\hat{}\sigma$ and $p_0\hat{}\tau$ which differ at only one number (which then must be a coding location $\langle k, n\rangle$ for some $k \in F_p$). Suppose σ, τ are strings as above with $|\sigma| = |\tau| = \ell$. Let $\sigma = \gamma_0^0, \gamma_1^0, \ldots, \gamma_z^0 = \tau$ be a list of strings in $\{0,1\}^\ell$ such that $\gamma_i^0, \gamma_{i+1}^0$ differ at only one number for each i. Let β be such that $\Phi_e^{X\oplus(p_0\hat{}\gamma_1^0\hat{}\beta)}(n)\downarrow$ (such a β exists by the same argument as before). Set $\gamma_i^1 = \gamma_i^0\hat{}\beta$ for each $0 \leq i \leq z$. Repeat this process for each $j \leq z$. At step $j+1$, let β be such that $\Phi_e^{X\oplus(p_0\hat{}\gamma_{j+1}^j\hat{}\beta)}(n)\downarrow$, and set $\gamma_i^{j+1} = \gamma_i^j\hat{}\beta$ for each $0 \leq i \leq z$. At the end, we have strings $\gamma_0^z, \gamma_1^z, \ldots, \gamma_z^z$ such that $\Phi_e^{X\oplus(p_0\hat{}\gamma_i^z)}(n)\downarrow$ for each i, and $p_0\hat{}\gamma_i^z, p_0\hat{}\gamma_{i+1}^z$ differ at only one number for each i. Since

$$\Phi_e^{X\oplus(p_0\hat{}\gamma_0^z)}(n) = \Phi_e^{X\oplus(p_0\hat{}\sigma)}(n) \neq \Phi_e^{X\oplus(p_0\hat{}\tau)}(n) = \Phi_e^{X\oplus(p_0\hat{}\gamma_z^z)}(n),$$

there must be an i for which $\Phi_e^{X\oplus(p_0\hat{}\gamma_i^z)}(n) \neq \Phi_e^{X\oplus(p_0\hat{}\gamma_{i+1}^z)}(n)$. The strings $p_0\hat{}\gamma_i^z, p_0\hat{}\gamma_{i+1}^z$ differ at only one number and it must be a coding location $\langle k, m\rangle$ for some $k \in F_p$ as required.

Next, we show that X can find infinitely many coding locations $\langle k, m\rangle$ for some fixed $k \in F_p$. Suppose we want to find such a location $\langle k, m\rangle$ with $m > M$. Search for strings $p_0\hat{}\sigma$ and $p_0\hat{}\tau$ that agree on the first M positions, differ at only one position, and satisfy $\Phi_e^{X\oplus(p_0\hat{}\sigma)}(n) \neq \Phi_e^{X\oplus(p_0\hat{}\tau)}(n)$. Such strings must exist because we could have started the above analysis at any condition $q \in \mathcal{G}$ with $q \leq p$ (so we can find such strings agreeing on arbitrarily long initial segments). The position at which $p_0\hat{}\sigma$ and $p_0\hat{}\tau$ differ must be a coding location bigger than M. Since F_p is finite, infinitely many of these coding locations must be for the same k. Given this k, X can find infinitely many coding locations $\langle k, c_{k,m}\rangle$. Hence, X can enumerate an infinite subset of C_k and so can compute C_k by our initial assumption on the C_i. ■

As 2-genericity sufficed for the proof of the theorem above , we can get the required $G_j \leq_T C''$ and, indeed with $(G_0 \oplus G_1)'' \equiv_T C''$. We show below (Theorem sw01 7.3.1 and Exercise sw1gen 7.3.3) that we can do better.

Now we work toward coding arbitrary countable relations on $\mathcal{D}$.

joinw1gen **Proposition 7.2.2** *If H is Cohen 1-generic over C, then, for any $i, j \in \omega$ and $X, Y \leq C$, if $X \oplus H^{[i]} \leq Y \oplus H^{[j]}$ then $i = j$ and $X \leq Y$.*

Proof. Suppose that for some e, $X, Y \leq_T C$, $\Phi_e^{Y\oplus H^{[j]}} = X \oplus H^{[i]}$ and consider the set

$$S_e = \{\sigma \in 2^{<\omega} : \exists n(\Phi_e^{Y\oplus\sigma^{[j]}}(n)\downarrow\neq X \oplus \sigma^{[i]}(n))\}.$$

$S_e \in \Sigma_1(C)$ so either there is a $\sigma \in S_e \cap H$ or there is a $\sigma \subset H$ no extension of which is in S_e. The first alternative clearly violates our assumption that $\Phi_e^{Y\oplus H^{[j]}} = X \oplus H^{[i]}$ and so there is a $\sigma \subset H$ such that $\tau \notin S_e$ for all $\tau \supseteq \sigma$. Let $n = |\sigma^{[i]}|$. If $i \neq j$ and there were $\beta \supseteq \sigma^{[j]}$ such that $\Phi_e^{Y \oplus \beta}(2n+1)\downarrow$, we could extend σ to a τ such that $\tau^{[j]} = \beta$ and $\tau^{[i]}(n) = 1 - \Phi_e^{Y \oplus \beta}(2n+1)$ (as the value of $\tau^{[i]}(n)$ is independent of $\tau^{[j]}$). In this case, we have

$$\Phi_e^{Y \oplus \tau^{[j]}}(2n+1)\downarrow \neq \tau^{[i]}(n) = (X \oplus \tau^{[i]})(2n+1)$$

and so $\tau \in S_e$, contradicting our choice of σ. Therefore, there can be no $\beta \supseteq \sigma^{[j]}$ making $\Phi_e^{Y \oplus \beta}(2n+1)$ converge while $\Phi_e^{Y \oplus H^{[j]}}$ is total by assumption and $\sigma^{[j]} \subset H^{[j]}$ for a contradiction. Thus $i = j$.

Next, we show that $X \leq_T Y$. To compute $X(n)$ from Y, search for a $\tau \supseteq \sigma$ such that $\Phi_e^{Y \oplus \tau^{[j]}}(2n)$ converges (such a τ exists because $\Phi_e^{Y \oplus H^{[i]}}$ is total and $\sigma^{[j]} \subset H^{[j]}$). Then, as usual, we claim that $\Phi_e^{Y \oplus \tau^{[j]}} = (X \oplus \tau^{[i]})(2n) = X(n)$ for if not, $\tau \in S_e$ and extends σ for a contradiction. ■

reldef **Theorem 7.2.3** *Every countable relation $R(x_0, \ldots, x_{n-1})$ on $\mathcal{D}$ is definable from parameters. Indeed, if C is a uniform upper bound on representatives C_i of the sets with degrees $\mathbf{c}_i$ in the domain of R as well as of the $\left\langle C_{j_0}, \ldots, C_{j_{n-1}} \right\rangle$ such that $R(\mathbf{c}_{j_0}, \ldots, \mathbf{c}_{j_{n-1}})$ and H is Cohen 1-generic over C then there is a notion of forcing recursive in $C \oplus H$ such that any 2-generic computes the required parameters. Moreover, for each n there is a formula $\varphi_n(x_0, \ldots, x_{n-1}, \bar{y})$ with $\bar{y}$ of length some $k > 0$ (depending only on n) which includes the clauses that $x_i \leq y_0$ for each $i < n$ such that as $\bar{\mathbf{p}}$ ranges over all k-tuples of degrees, the sets of n-tuples of degrees $\{\bar{\mathbf{a}} | \mathcal{D} \vDash \varphi(\bar{\mathbf{a}}, \bar{\mathbf{p}})\}$ range over all countable n-ary relations on $\mathcal{D}$.*

Proof. We take $\mathbf{c} = \deg(C)$ to be our first parameter. Let H be Cohen 1-generic over C and $\mathbf{h}_{i,j}$ be the degree of $H^{[\langle i,j \rangle]}$. We code R using the following countable sets of pairwise incomparable degrees.

$$\mathcal{H}_i = \{\mathbf{h}_{i,j} | j \in \mathbb{N}\} \text{ for } i < n$$

$$\mathcal{F}_i = \{\mathbf{c}_j \vee \mathbf{h}_{i,j} | j \in \mathbb{N}\} \text{ for } i < n$$

$$\mathcal{R} = \{\mathbf{h}_{0,j_0} \vee \mathbf{h}_{1,j_1} \vee \cdots \vee \mathbf{h}_{n-1,j_{n-1}} : R(\mathbf{c}_{j_0}, \mathbf{c}_{j_1}, \ldots, \mathbf{c}_{j_{n-1}})\}$$

Each of these sets consists of pairwise incomparable degrees. The first and third by Proposition genindep 6.2.23 that for a Cohen 1-generic H the sets $H^{[k]}$ form a very independent set. (So, for any finite sets A and B of $\mathbf{h}_{i,j}$, $\vee\{\mathbf{x}|\mathbf{x} \in A\} \leq \vee\{\mathbf{x}|\mathbf{x} \in B\}$ if and only if $A \subseteq B$.) The elements of each $\mathcal{F}_i$ are pairwise incomparable by Proposition joinw1gen 7.2.2. Our defining formula φ for R is now

$$\&_{i<n}(\mathbf{x}_i \leq \mathbf{c}) \;\&\; (\exists \mathbf{y}_i)_{i<n}(\mathbf{y}_i \in \mathcal{H}_i \;\&\; \&_{i<n}(\mathbf{x}_i \vee \mathbf{y}_i) \in \mathcal{F}_i \;\&\; \vee\{\mathbf{y}_i | i < n\} \in \mathcal{R})$$

where we understand membership in the sets $\mathcal{H}_i$, $\mathcal{F}_i$ and $\mathcal{R}$ as being defined by the appropriate formulas and parameters as given by Theorem sw 7.2.1. This also supplies the notion of forcing required in our Theorem by taking (the disjoint union of) three versions of the one provided in Theorem 7.2.1 for the three families of pairwise Turing incomparable sets needed for these definitions as they are uniformly recursive in $C \oplus H$. The verification that this formula defines the relation is straightforward. If $R(\bar{\mathbf{x}})$ then every element of the sequence $\bar{\mathbf{x}}$ is below $\mathbf{c}$ and is therefore equal to a $\mathbf{c}_{j_i}$ (for $i < n$). The degrees $\mathbf{h}_{i,j_i} \in \mathcal{H}_i$ then are the witness $\mathbf{y}_i$ required in φ. In the other direction, if φ holds of any n-tuple then all its elements are below $\mathbf{c}$ and we need to consider the situation where $\varphi(\mathbf{x}_{j_0}, \ldots \mathbf{x}_{j_{n-1}})$ for some j_i, $i < n$. Let the required witnesses be $\mathbf{y}_i$. As $\mathbf{y}_i \in \mathcal{H}_i$ and $(\mathbf{x}_{j_i} \vee \mathbf{y}_i) \in \mathcal{F}_i$, $\mathbf{y}_i = \mathbf{h}_{i,j}$. Then as $\bigvee_{i<n} \mathbf{y}_i \in \mathcal{R}$, $R(\mathbf{x}_{j_0}, \mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_{n-1}})$. The assertions in the Theorem about the form of the required formulas φ are now immediate from Theorem 7.2.1. ■

Note that with the above assumptions on $\mathbf{c}$ in this proof, Theorem 7.2.1, the remarks immediately following it and Proposition 6.2.13, we can get all the parameters need for this definition of R below $\mathbf{c}''$. We improve this by one jump in the next section.

We can now precisely characterize the complexity of $Th(\mathcal{D})$ as that of true second order arithmetic.

ThD **Theorem 7.2.4** *$Th(\mathcal{D}, \leq) \equiv_1 Th^2(\mathbb{N}, \leq, +, \times, 0, 1)$.*

Proof. That $Th(\mathcal{D}, \leq) \leq_1 Th^2(\mathbb{N}, \leq, +, \times, 0, 1)$ is easy. As $A \leq_T B$ is definable in arithmetic (indeed as we have seen it is Σ_3 in A and B) and quantification over all sets gives quantification over all degree, we can recursively translate any sentence about $\mathcal{D}$ to an equivalent one of about second order arithmetic. For the other direction we use the formulas φ_1, φ_2 and φ_3 of Theorem 7.2.3 to give an interpretation of true second order arithmetic in $\mathcal{D}$. We consider sequences of parameters $\bar{p}_D$, $\bar{p}_+$, $\bar{p}_\times$ and $\bar{p}_<$ so that $\varphi_1(\bar{p}_D)$ defines a countable set of degrees and plays the role of φ_D for our interpretation. Our correctness condition then includes the sentences that say that $\varphi_3(\bar{p}_+)$, $\varphi_3(\bar{p}_\times)$ and $\varphi_2(\bar{p}_<)$ (playing the roles of φ_+, $\varphi_\times$ and $\varphi_<$, respectively) define relations on the countable set defined by $\varphi_1(\bar{p}_D)$ to determine a structure $\mathcal{M}(\bar{p})$ (where $\bar{p}$ is the concatenation of all the sequences of parameters used here) that satisfies all the axioms of our finite theory of arithmetic. Theorem 7.2.3 then says that there are choices of these parameters such that the structure so defined is isomorphic to $\mathbb{N}$. After all, $\mathbb{N}$ is just a countable set with two ternary relations and one binary one. We now use $\varphi_1(x, \bar{q}) \wedge \varphi_1(\bar{p}_D)$ as the φ_S required for our interpretation of true second order arithmetic. Again by Theorem 7.2.3, as $\bar{q}$ ranges over tuples of degrees, the subsets of $M(\bar{p})$ defined by φ_s range over all subsets of $M(\bar{p})$ as required. All that remains to do is to show that we can extend the list of correctness conditions that guarantee that $\mathcal{M}(\bar{p})$ is a model of our finite axiomatization of arithmetic to also guarantee that it is isomorphic to $\mathbb{N}$. We can do this by adding on the sentence which asserts that every nonempty subset of $M(\bar{p})$ (as given by $\varphi_S(\bar{q}, \bar{p})$ for some $\bar{q}$) has an $<_{\mathcal{M}}$ least element, i.e. $\forall \bar{q}\{\exists x(\varphi_S(x, \bar{q}, \bar{p})) \rightarrow \exists x[\varphi_S(x, \bar{q}, \bar{p}) \wedge \neg\exists y(\varphi_S(y, \bar{q}, \bar{p}) \wedge \varphi_<(y, x, \bar{p}_<))]\}$.
■

thjumpideal **Exercise 7.2.5** *If $\mathcal{C}$ is a jump ideal of $\mathcal{D}$ (i.e. a downward closed subset that is also closed under jump and join), then the theory of $\mathcal{C}$ is 1-1 equivalent to that of the model of second order arithmetic where set quantifiers range over the sets with degrees in $\mathcal{C}$.*

Notes: Slaman and Woodin forcing was introduced in Slaman and Woodin [1986] where they proved Theorems $\overset{\texttt{sw}}{7.2.1}$ and $\overset{\texttt{reldef}}{7.2.3}$. Theorem $\overset{\texttt{ThD}}{7.2.4}$ (which as presented here follows easily from these results) is originally due to Simpson [1977] although with a very different proof using then new initial segments results and Theorem $\overset{\texttt{exactpair}}{5.2.14}$. Another version using simpler codings and previously know initial segment results along with Theorem $\overset{\texttt{exactpair}}{5.2.14}$ is in Nerode and Shore [1980]. Exercise $\overset{\texttt{thjumpideal}}{7.2.5}$ is from Nerode and Shore [1980a].

7.3 $Th(D \leq 0')$ ThD<0'

We now want to improve our coding results so that they become applicable below $\mathbf{0}'$. We begin with the Slaman and Woodin coding of sets of pairwise incomparable degrees.

sw0' **Theorem 7.3.1** *For any set $S = \{C_0, C_1, \ldots\}$ of pairwise Turing incomparable subsets of $\mathbb{N}$ let $C = \oplus C_i$. There are then G_0,$G_1 \leq_T C'$ and D_i such that, for every $i \in \mathbb{N}$ and $j < 2$,, $D_i \leq_T C_i \oplus G_j$ while $D_i \not\leq_T C_i$. Moreover, the C_i are minimal with this property among sets recursive in C in the sense that for any $X \leq_T C$ for which there is a D such that $D \leq_T X \oplus G_j$ $(j < 2)$ but $D \not\leq_T X$ there is an i such that $C_i \leq_T X$.*

Proof. We follow the ideas of the proof of Theorem $\overset{\texttt{sw}}{7.2.1}$ but replace the uses of 2-genericity for extending conditions to make something converge. At various steps we ask if there are appropriate extensions, if so we take them and continue our construction. If not we have a condition that forces some functional to diverge and so can satisfy the relevant requirement in that way. ■

Proof. We build $D_i \leq_T G_0 \oplus C_i, G_1 \oplus C_i$ such that $D_i \not\leq_T C_i$. The requirements for diagonalization here are:

$$P_{e,i} : \Phi_e^{C_i} \neq D_i.$$

Let $X_j = \Phi_j^C$. We also have requirements for minimality:

$$R_{e,,j} : \Phi_e^{G_0 \oplus X_j} = \Phi_e^{G_1 \oplus X_j} = D \Rightarrow D \leq_T X_j \text{ or } \exists i (C_i \leq_T X_j).$$

We list all the requirements as Q_s. We build G_0, G_1 by finite approximations $\gamma_{0,s}, \gamma_{1,s}$ of equal length. As before we let $D_i(m) = G_0(\langle i, c_{i,m} \rangle)$ where $\{c_{i,m}\}$ is an enumeration of C_i in increasing order. So $D_i \leq_T G_0 \oplus C_i$. We guarantee that $D_i \leq_T G_1 \oplus C_i$ as before by making sure that, for each i, $G_0(\langle i, c_{i,m}\rangle \neq G_1(\langle i, c_{i,m}\rangle)$ for at most finitely many m. In particular we institute a *rule for the construction* that when we act to satisfy requirement Q_n at stage s by extending the current values of γ_k $(k = 0, 1)$ we require, for $i \leq n$, $\langle i, m \rangle \geq |\gamma_{0,s}| = |\gamma_{1,s}|$ and $m \in C_i$, that the extensions γ'_k are such that $\gamma'_0(\langle i, m \rangle) = \gamma'_1(\langle i, m \rangle)$. As we act to satisfy any Q_n at most once, this rule guarantees that there are at most finitely many relevant differences between G_0 and G_1 for each i.

At stage s, if $Q_s = P_{e,i}$, we act to satisfy $P_{e,i}$. Choose m such that $\langle i, c_{i,m} \rangle \geq |\gamma_{0,s}|$. Ask if $\Phi_e^{C_i}(m) \downarrow$. If not, let $\gamma_{k,s+1} = \gamma_{k,s}$ for $k = 0, 1$. (As usual this satisfies $P_{e,i}$.) If it does converge, extend each of $\gamma_{0,s}, \gamma_{1,s}$ by the same string σ to $\gamma_{0,s+1}, \gamma_{1,s+1}$ with $\gamma_{0,s+1}(\langle i, c_{i,m} \rangle) \neq \Phi_e^{C_i}(m)$. This also satisfies the requirement because $D_i(m) = G_0(\langle i, c_{i,m} \rangle)$ by definition and trivially obeys the rule of the construction.

Note that C' can decide if $\Phi_e^{C_i}(m) \downarrow$, so this action is recursive in C'.

If $Q_s = R_{e,j}$, this stage has a substage for each requirement $Q_n = R_{e',j'}$ with $n \leq s$ that has not yet been satisfied. For notational convenience we write γ_k for $\gamma_{k,s}$ in the description of our action at stage s. At the end of each substage we define successive extensions $\gamma_{k,l}$ of γ_k satisfying the rule of the construction. We first try to satisfy $R_{e,j}$ (which, of course, we have not attempted to satisfy before). We ask if $\exists x \exists \sigma_k \supseteq \gamma_k$ which satisfy the rule of our construction and such that the $\sigma_k \oplus X$ e-split at x, i.e.

$$\Phi_e^{\sigma_0 \oplus X_j}(x) \downarrow \neq \Phi_e^{\sigma_1 \oplus X_j}(x) \downarrow .$$

Note that, when we are acting to satisfy any Q_n, checking if extensions of the current values of γ_k satisfy the rule of the construction is recursive in $\oplus\{C_i | i \leq n\}$ and so uniformly recursive in C. Thus this question can be answered by C'. There is one subtlety here. We must be careful with what we mean by a computation from X_j as there is no list of all the sets recursive in C that is uniformly recursive in C. So what we mean here is that there is a computation of Φ_j^C providing a long enough initial segment of X_j so as to make the desired computations at m converge. This makes the whole question one that is Σ_1^C and so recursive in C'.

If the answer is yes, choose as usual the first such extensions (in a uniform search recursive in C) as $\gamma_{0,0}, \gamma_{1,1}$. Note that we have now satisfied $R_{e,j}$. If the answer is no, ask if $\exists x \exists \sigma, \tau \ \ ((\gamma_0\hat{\ }\sigma \oplus X_j)|_e(\gamma_0\hat{\ }\tau \oplus X))$ (See Definition esplit 5.2.10). This question is also $\Sigma_1(C)$.

- If not, let $\gamma_{k,s,0} = \gamma_{k,s}$. Then, as usual, if $\Phi_e^{G_0 \oplus X_j}$ is total, it is recursive in X as we guarantee that $G_0 \supseteq \gamma_{0,0}$. To calculate it at x, find any σ such that $\Phi_e^{\gamma_0\hat{\ }\sigma \oplus X_j}(x) \downarrow$. This computation must give right answer. So in this case we have also satisfied $R_{e,j}$.

- If so,we can find such σ and τ (recursively in C). We interpolate between σ, τ with strings $\sigma = \delta_0 = \delta_1, \ldots, \delta_z = \tau$ which differ successively at exactly one number. Ask if $\exists \sigma_1$ such that $\Phi_e^{\gamma_0\hat{\ }\delta_1\hat{\ }\sigma_1 \oplus X_j}(x) \downarrow$. If not, let $\gamma_{k,0} = \gamma_k\hat{\ }\delta_1$. Note that this extension satisfies the rule of the construction and that we have satisfied $R_{e,j}$ by guaranteeing that $\Phi_e^{G_0 \oplus X_j}(x) \uparrow$. If yes, consider $\delta_2\hat{\ }\sigma_1$ and ask again if there is a σ_2 such that $\Phi_e^{\delta_2\hat{\ }\sigma_1\hat{\ }\sigma_2 \oplus X_j}(x) \downarrow$. If not, let $\gamma_{k,0} = \delta_2\hat{\ }\sigma_1$ as before obeying the rule of the construction and satisfying $R_{e,j}$. If so, we continue on inductively through the δ_k.

- Eventually we either define $\gamma_{k,0}$ and satisfy $R_{e,j}$ or we find $\sigma_1, \ldots, \sigma_z$ such that $\Phi_e^{\gamma_0\hat{}\delta_l\hat{}\rho\oplus X_j}(x)\downarrow$ for every $l \leq z$ where $\rho = \sigma_1\hat{}\ldots\hat{}\sigma_z$. In the second case, we set $\gamma_{k,0} = \gamma_{k,s}$. This action does not satisfy $R_{e,i}$ but it demonstrates that there are $\hat{\sigma}$ and $\hat{\tau}$ which differ at exactly one number and for which $(\gamma_0\hat{}\hat{\sigma} \oplus X)|_e(\gamma_0\hat{}\hat{\tau} \oplus X)$. The point here is that, as $\Phi_e^{\gamma_0\hat{}\delta_0\hat{}\rho\oplus X_j}(x)\downarrow = \Phi_e^{\gamma_0\hat{}\sigma\oplus X_j}(x)\downarrow \neq \Phi_e^{\gamma_0\hat{}\tau\oplus X_j}(x)\downarrow = \Phi_e^{\gamma_0\hat{}\delta_z\hat{}\varepsilon\oplus X_j}(x)\downarrow$, there is an l such that $\Phi_e^{\gamma_0\hat{}\delta_l\hat{}\rho\oplus X_j}(x)\downarrow \neq \Phi_e^{\gamma_0\hat{}\delta_{l+1}\hat{}\rho\oplus X_j}(x)\downarrow$ while $\delta_l\hat{}\rho$ and $\delta_{l+1}\hat{}\rho$ differ at exactly one number. Now consider $\gamma_1\hat{}\hat{\sigma}$. If there is no μ such that $\Phi_e^{\gamma_1\hat{}\hat{\sigma}\hat{}\mu\oplus X_j}(x)\downarrow$ then we can again satisfy $R_{e,j}$ by setting $\gamma_{k,s,0} = \gamma_{k,s}\hat{}\hat{\sigma}$. If there is such a μ, we compare $\Phi_e^{\gamma_1\hat{}\hat{\sigma}\hat{}\mu\oplus X_j}(x)\downarrow$ with $\Phi_e^{\gamma_0\hat{}\hat{\sigma}\hat{}\mu\oplus X_j}(x)\downarrow$ and $\Phi_e^{\gamma_0\hat{}\hat{\tau}\hat{}\mu\oplus X_j}(x)\downarrow$. As the last two are different one of them must be different from the first. If $\Phi_e^{\gamma_1\hat{}\hat{\sigma}\hat{}\mu\oplus X_j}(x)\downarrow \neq \Phi_e^{\gamma_0\hat{}\hat{\sigma}\hat{}\mu\oplus X_j}(x)\downarrow$, we would contradict our assumption that the answer to our very first question was no as $\gamma_1\hat{}\hat{\sigma}\hat{}\mu$ and $\gamma_0\hat{}\hat{\sigma}\hat{}\mu$ certainly satisfy the rule of the construction. If $\Phi_e^{\gamma_1\hat{}\hat{\sigma}\hat{}\mu\oplus X_j}(x)\downarrow \neq \Phi_e^{\gamma_0\hat{}\hat{\tau}\hat{}\mu\oplus X_j}(x)\downarrow$, the only way we would not have the same contradiction is if the one point at which $\hat{\sigma}$ and $\hat{\tau}$ differ is a coding location $\langle k, c_{k,m}\rangle$ with $k < s$. Thus the only way our actions at this stage do not satisfy $R_{\langle e,j\rangle}$ is if there are $\hat{\sigma}\hat{}\mu$ and $\hat{\tau}\hat{}\mu$ which differ at at exactly one point such that $(\gamma_1\hat{}\hat{\sigma}\hat{}\mu \oplus X_j)|_e(\gamma_0\hat{}\hat{\tau}\hat{}\mu \oplus X_j)$ and for any such $\hat{\sigma}$ and $\hat{\tau}$ the point of difference must be a coding location $\langle k, c_{k,m}\rangle$ with $k < s$.

- In this last case we set $\gamma_{0,0} = \gamma_0$ and $\gamma_{1,0} = \gamma_{1,s}$. In any event, we now proceed to extend $\gamma_{1,0}$ (and then γ_1) in the same way but attempting to satisfy each $Q_n = R_{e',j'}$ with $n < s$ that has not yet been satisfied. After some finite number of such attempts we have tried them all, satisfying some and for the others producing one more example of an x and two strings $\hat{\sigma}$ and $\hat{\tau}$ differing at one number only (after $|\gamma_0|$) such that $(\gamma_0\hat{}\hat{\sigma} \oplus X_{j'})|_e(\gamma_1\hat{}\hat{\tau} \oplus X_{j'})$ for each $\langle e', j'\rangle$ which we have not yet satisfied and a guarantee that any two such strings differ at a coding location $\langle k, c_{k,m}\rangle$ with $k < n$.

- At the end of this process we let $\gamma_{k,s+1}$ be the final extension of γ_k that we have produced.

We now claim that all the requirements are satisfied. It is immediate that $P_{e,i}$ is satisfied when we act for $Q_s = P_{e,i}$ at stage s. Consider any $R_{e,j} = Q_{s_0}$. If we ever act so as to satisfy it at some stage s of the construction, it is clearly satisfied and we never act for it again. As we violate the rule of the construction at some $\langle k, c_{k,m}\rangle$ only when we act to satisfy requirement Q_n for $n \leq k$ and we do so at most once for each n, $D_i \leq_T G_1 \oplus C_i$ as required.

Finally, suppose that the first requirement that we never act to satisfy during the construction is Q_n. It must be some $R_{e,j}$. Suppose that all requirements Q_r for $r < n$ have been satisfied by stage $s_0 > n$. At each stage $s > s_0$ with $Q_s = R_{e',j'}$ we attempt to satisfy $R_{e,j}$ at some substage of the construction. As we fail, there are $\Phi_{e'}^{\delta_0\oplus X_{j'}}(x)\downarrow \neq \Phi_{e'}^{\delta_1\oplus X_{j'}}(x)\downarrow$ with $\delta_k \supseteq \gamma_{k,s} \supseteq \gamma_{k,n}$ which differ at exactly one point and any such pair differ at a coding location $\langle k, c_{m,k}\rangle$ with $k \leq n$. Recursively in X_j we can then search for and find infinitely many extensions δ_k of $\gamma_{k,n}$ with this property with the points at which they differ becoming arbitrarily large (as $|\gamma_{k,s}|$ is clearly going to infinity). As there are only finitely many $k \leq n$, there must be one $k \leq n$ for which infinitely many of these δ_k differ at a point of the form $\langle k, z\rangle$ with infinitely many different z. As every such point is a coding location, recursively in X we can compute an infinite subset of C_k, so by our initial assumption that each C_i is recursive in everyone of its infinite subsets $C_k \leq_T X_j$ as required for $R_{e,j}$ to be satisfied in the end. ■

This step-by-step construction is the much the same as the forcing argument we saw before, but grittier, and we gain a quantifier. This helps us determine the true complexity of $Th(\mathcal{D}, \leq \mathbf{0}')$: $Th(\mathcal{D}, \leq \mathbf{0}') \equiv_m Th(\mathbb{N}, +, \times, \leq)$.

Exercise 7.3.2 *It is easy to show that the G_i of Theorem sw0' 7.3.1 can be made to have (or already have) jumps below C'.??Need this for definability in $\mathcal{D}(\leq\mathbf{0}')$ do proof or some details??*

sw1gen **Exercise 7.3.3** *With the notation as in Theorem sw 7.2.1 show that for any $\mathcal{G}$ 1-generic for $\mathcal{P}$, G_0 and G_1 have the properties required by the Theorem. So in particular, we can make $G_0' \equiv_T 0' \equiv_T G_1'$. This then supplies the analogous result for Theorem reldef 7.2.3, i.e. a notion of forcing recursive in the appropriate $C \oplus H$ such that any 1-generic computes the parameters necessary to define the given relation. Hint: This is not easy. A proof can be found in Greenberg and Montalbán [2003]. ??Do out, need just for minimality 205-207????$(G_0 \oplus G_1)' \equiv_T 0'$??.*

reldef0' **Theorem 7.3.4** *If R is an n-ary relation on $\mathcal{D}(\leq\mathbf{0}')$ which is uniformly recursive in a low degree $\mathbf{c}$ in the sense that there are families of sets $\{X_i\} = S$ and $\{\langle X_{i_1}, \ldots, X_{i_n}\rangle\} = T$ uniformly recursive in $C \in \mathbf{c}$ such that $\{\deg(X_i) | X_i \in S\}$ is the field of R (i.e. all elements that occur in any n-tuple satisfying R) and $\{\langle \deg X_{i_1}, \ldots, \deg X_{i_n}\rangle \mid \langle X_{i_1}, \ldots X_{i_n}\rangle \in T\} = R$, then there are $\bar{\mathbf{p}} < \mathbf{c}' = \mathbf{0}'$ which define R by the formula φ_n of Theorem reldef 7.2.3.*

Proof. We begin with a G which is Cohen 1-generic over C so that $(C \oplus G)' \equiv_T C'$. The set of degrees $\mathcal{R}$ and the finite families of sets of degrees $\mathcal{H}_i$ and $\mathcal{F}_i$ of the proof of Theorem reldef 7.2.3 are all now uniformly recursive in $C \oplus G$ and consist of pairwise Turing incomparable sets so, by Theorem sw0' 7.3.1, there are sequences of parameters defining each of them all below $(C \oplus G)'$. The proof of Theorem reldef 7.2.3 now shows that they define R as required. ■

We now explain how we plan to code arithmetic in $\mathcal{D}(\leq 0')$. The "intended model" starts with an nice effective successor structure determined by parameters $\bar{\mathbf{q}}$: $\mathbf{c}$, $\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{e}_0$, $\mathbf{e}_1$, $\mathbf{d}_0$, $\mathbf{f}_0$ and $\mathbf{f}_1$ with $\mathbf{c}' = 0'$ and $\mathbf{c}$ being above all of the other parameters and all the required $\hat{\mathbf{d}}_n$ as well. Moreover, the $\mathbf{d}_n$ are all uniformly recursive in $\mathbf{c}$. We can do this by Exercise latemb0' 6.3.15 or latemb1gen ??. We then choose, as in the proof of Theorem ThD 7.2.4 parameters $\bar{\mathbf{p}}_D$, $\bar{\mathbf{p}}_+$, $\bar{\mathbf{p}}_\times$ and $\bar{\mathbf{p}}_<$ so that $\varphi_1(\bar{\mathbf{p}}_D)$ defines $\{\mathbf{d}_n | n \in \mathbb{N}\}$ and $\varphi_3(\bar{\mathbf{p}}_+)$, $\varphi_3(\bar{\mathbf{p}}_\times)$ and $\varphi_2(\bar{\mathbf{p}}_<)$ (playing the roles of φ_+, $\varphi_\times$ and $\varphi_<$, respectively) that define relations on the countable set defined by $\varphi_1(\bar{\mathbf{p}}_D)$ to determine a structure $\mathcal{M}(\bar{p})$ (where $\bar{\mathbf{p}}$ is the concatenation of all the sequences of parameters used beginning with $\bar{\mathbf{q}}$) that satisfies all the axioms of our finite theory of arithmetic and such that $\mathbf{d}_0$ is the least element in the ordering of $\mathcal{M}(\bar{\mathbf{p}})$ given by $\varphi_2(\bar{\mathbf{p}}_<)$ and, for each n, $\mathbf{d}_{n+1}$ is the immediate successor of $\mathbf{d}_n$ in this order. We can find such parameters below $\mathbf{0}'$ by the arguments for the proof Theorem 7.2.3 combined with Theorem 7.3.1 (relativized to $\mathbf{c}$) since the $\mathbf{d}_n$ and the desired relations on them are uniformly recursive in $\mathbf{c}$ and $\mathbf{c}' = \mathbf{0}'$. Now this model is standard since the $\mathbf{d}_n$ are ordered in order type ω and constitute the universe of the model.

The problem is that there is no obvious way to definably say that the universe of the model is precisely the $\mathbf{d}_n$ in terms of just the prescribed parameters (or any other finite list). The issue is that we only have a scheme to generate these degrees not one to define them. We can come fairly close in a first order way. In addition to the correctness conditions that guarantee that the defined relations give a model of arithmetic on $\{x|\varphi_D(x, \bar{\mathbf{p}})\}$, we can approximate niceness by adding the sentences $\mathbf{c} \not\geq \mathbf{b}$ and $\forall d[\varphi_D(d) \rightarrow d \vee \mathbf{c} \geq \mathbf{b}\ \&\ \exists \hat{d}(d \wedge \hat{d} = 0\ \&\ (\forall d^* \neq d)(\varphi_D(d^*) \rightarrow (d \wedge d^* = \mathbf{0})\ \&\ (\hat{d} \geq d^*))]$. We can approximate the desired condition that $\{\mathbf{d}_n|n \in \omega\}$ is the domain of our structure by saying that $\mathbf{d}_0$ is the least element in the ordering of $\mathcal{M}(\bar{\mathbf{p}})$ given by $\varphi_2(\bar{\mathbf{p}}_<)$ and for every $\mathbf{d}$ such that $\varphi_D(\mathbf{d}, \bar{\mathbf{p}})$, if $\mathbf{d}$ is an even number in $\mathcal{M}(\bar{\mathbf{p}})$, then $(\mathbf{e}_0 \vee \mathbf{d}) \wedge \mathbf{f}_0$ is its immediate successor in the ordering given by $\varphi_2(\bar{\mathbf{p}}_<)$ while if it is an odd number then its immediate successor is given by $(\mathbf{e}_1 \vee \mathbf{d}) \wedge \mathbf{f}_1$. This guarantees that $\{\mathbf{d}_n|n \in \omega\}$ is the standard part of the model $\mathcal{M}(\bar{\mathbf{p}})$. Thus if we had a formula $\hat{\varphi}_S(x, \bar{r}, \bar{\mathbf{p}})$ which, as $\bar{r}$ ranged over n-tuples from $\mathcal{D}(\leq\mathbf{0}')$, defined a collection of subsets of $\mathcal{M}(\bar{\mathbf{p}})$ that include $\{\mathbf{d}_n|n \in \omega\}$, we could guarantee that $\mathcal{M}(\bar{P})$ was standard by saying that every subset (i.e. picked out by some choice of parameters $\bar{r}$) of $\mathcal{M}(\bar{\mathbf{p}})$ which contains its least element $(\mathbf{d}_0)$ and is closed under immediate successor is all of $\mathcal{M}(\bar{\mathbf{p}})$.

The crucial point now is that the proof of Proposition 6.4.9 shows that, under these conditions, $\{\mathbf{d}_n|n \in \omega\} \in \Sigma_3^C$ as is the ideal generated by this set. That is, the standard part of any $\mathcal{M}(\bar{\mathbf{p}})$ for $\bar{\mathbf{p}}$ satisfying all of these correctness conditions and the ideal it generates are both Σ_3^C. Our goal now is to prove that for every $\mathbf{c} < \mathbf{0}'$ and every Σ_3^C ideal in the degrees below $\mathbf{c}$, there are $\mathbf{g}_0, \mathbf{g}_1 \leq_T \mathbf{0}'$ which are an exact pair for the given ideal. Proposition 6.4.9 and Remark ?? then show that we could define the desired set $\{\mathbf{d}_n|n \in \omega\}$ in terms of this exact pair. We later prove this required result as Theorem 8.2.11. It supplies the final ingredient of our theorem.

Th(D<0') **Theorem 7.3.5** *$Th(\mathcal{D} \leq\mathbf{0}') \equiv_{1-1} Th(\mathbb{N})$.*

Proof. The above argument (together with Theorem 8.2.11) shows that we can interpret true first order arithmetic in $\mathcal{D}(\leq\mathbf{0}')$. Thus $Th(\mathbb{N}) \leq_{1-1} Th(\mathcal{D} \leq\mathbf{0}')$. The other direction is immediate since we can define the sets recursive in $0'$ in arithmetic as well as the ordering of Turing reducibility on them. Thus we have a recursive translation of sentences about $\mathcal{D}(\leq\mathbf{0}')$ to ones of arithmetic that preserves truth. Of course, this implies that $Th(\mathcal{D} \leq\mathbf{0}') \leq_{1-1} Th(\mathbb{N})$. ■

Notes: Theorem sw0' 7.3.1 and a special case of Theorem reldef0' 7.3.4 are in Slaman and Woodin [1986]. The full version of Theorem reldef0' 7.3.4 is in Odifreddi and Shore [1991] as is the proof of Theorem Th(D<0') 7.3.5 which is originally due to Shore [1981].

Chapter 8

Domination Properties

8.1 Introduction

An important topic in the study of the complexity of functions from $\mathbb{N}$ to $\mathbb{N}$ is the notion of rate of growth and of one function growing faster than another or faster than a whole class of functions. These issues are not only natural but they have important connections with the computational complexity of the functions as measured by Turing and other reducibilities. In this chapter we will study some of these notions and their impact on the structure of the degrees. They will play a crucial role in our analysis of the complexity of important degree structures including $D(\leq \mathbf{0}')$ which we study in this chapter ??and all of $\mathcal{D}$ as well as many jump ideals that we will study in later chapters??. We begin with some basic definitions.

Definition 8.1.1 *1. The function g* dominates *the function f ($f < g$) if, for all but finitely many x, $f(x) < g(x)$.*

2. The degree $\mathbf{g}$ dominates *the function f if some $g \in \mathbf{g}$ dominates f.*

3. The function g dominates *the degree $\mathbf{f}$ if g dominates every function $f \in \mathbf{f}$.*

4. The degree $\mathbf{g}$ dominates *the degree $\mathbf{f}$ if for every $f \in \mathbf{f}$ there is a $g \in \mathbf{g}$ which dominates f.*

*We also sometimes express these relations in the passive form saying, for example, that f is g-*dominated *or f is $\mathbf{g}$-*dominated *for the first two relations. A function g that dominates the degree $\mathbf{0}$ is called* dominant*.*

In the literature a degree $\mathbf{f}$ that is not $\mathbf{0}$-dominated (i.e. there is an $f \in \mathbf{f}$ which is not dominated by any recursive function) is, for historical reasons unrelated to our concerns, called *hyperimmune*. If $\mathbf{f}$ is not hyperimmune, i.e. it is $\mathbf{0}$-dominated, is also called *hyperimmune free*. For example, we show later that every $\mathbf{0} < \mathbf{a} < \mathbf{0}'$ is hyperimmune (Theorem delta2dom 8.2.3) while the minimal degrees constructed by Spector (§spectormin 9.2) are hyperimmune free.

0domtt **Exercise 8.1.2** *Prove that if* $\mathbf{a}$ *is* $\mathbf{0}$*-dominated and* $B \leq_T A \in \mathbf{a}$ *then* $B \leq_{tt} A$*. So any* $\mathbf{0}$*-dominated Turing degree consists of exactly one tt (and so wtt) degree. Hint: if* $B = \Phi_e^A$ *then consider the function* f *such that* $f(n) = \mu s(\Phi_{e,s}^{A\restriction s}(n)\downarrow)$*.*

8.2 R.E. and Δ_2^0 degrees

redom **Theorem 8.2.1** *If* $A >_T 0$ *is r.e. then there is a function* $m \equiv_T A$ *which is not* $\mathbf{0}$*-dominated, i.e. it is not dominated by any recursive function. Indeed, any function* g *which dominates* m *computes* A*.*

Proof. For A r.e., let A_s be the standard approximation to A at stage s. Let m be the least modulus function for this approximation: $m(x) = \mu s(\forall t \geq s)(A_s \restriction x = A_t \restriction x)$. For r.e. sets, the approximation changes its mind at most once and is correct in the limit, so $m(x)$ is also the $\mu s(A_s \restriction x = A \restriction x)$ and is clearly of the same degree as A. Moreover, if $g(x) \geq m(x)$ for almost all x, then $A \leq_T g$ as $A \restriction x = A_{g(x)} \restriction x$ for all but finitely many x. Thus, if $A >_T 0$, then m is not dominated by any recursive function and any g that dominates m computes A. ■

The Shoenfield limit lemma (Theorem limitlemma 4.3.9) gives us a recursive approximation $h(x,s)$ to any $A \in \Delta_2^0$ (or equivalently $A \leq_T 0'$). So the least modulus function m makes sense for such an approximation as well. So does the second version used in the above proof. Here we call it the *computation function:* $f(x) = \mu(s > x)(\forall y < x)(h(y,s) = A(y))$ (for technical reasons, we do not consider first few stages). It calculates the first stage after x at which the approximation is correct up to x. But, since we are no longer looking at r.e. sets, the approximation might change even after it's correct and the computation function f need not be the same as the least modulus m. The two functions may not be the same even up to degree.

Exercise 8.2.2 *Find an* $A <_T 0'$ *and an approximation* $h(x,s)$ *to* A *for which the least modulus function* m *computes* $0'$*. On the other hand, the computation function* f *for* h *is always of the same degree as* A*.*

We can, nonetheless extend Theorem redom 8.2.1 to all $A \in \Delta_2^0$.

delta2dom **Theorem 8.2.3** *If* A *is* Δ_2^0*, then there is an* $f \equiv_T A$ *which is not* $\mathbf{0}$*-dominated. Indeed, any function* g *which dominates* f *computes* $\mathbf{a}$*.*

Proof. By the Shoenfield limit lemma, there is a recursive $h(x,s)$ such that $\lim_{s\to\infty} h(x,s) = A(x)$. Let $f(x)$ be the computation function for this approximation. Suppose $f < g$. We claim that even though $h(z,s)$ may change at $z < x$ for $s > f(x)$, we can still compute A from g. Let s_0 be such that $(\forall m \geq s_0)(f(m) < g(m))$. To calculate $A(n)$ for $n > s_0$ find an $s > n$ such that $h(n,t)$ is constant for $t \in [g(s), gg(s)]$. Since $h(n,t)$ is eventually constant, such an s exists. Moreover, we can find it recursively in g: compute the intervals $[g(n+1), gg(n+1)], [g(n+2), gg(n+2)], [g(n+3), gg(n+3)], \ldots$ checking to see if

h is constant on the intervals. By the clause that makes $f(x) > x$ in the definition of the computation function and our choice of s_0, $gg(s) > fg(s) > g(s)$, so the first $t > g(s)$ at which h is correct for all elements below $g(s)$ is in $[g(s), gg(s)]$. For this t, $h(n,t) = A(n)$. As we chose s so that the value of $h(n,t)$ is constant on this interval, $A(n) = h(n,t)$ for any $t \in [g(s), gg(s)]$ and we have computed A recursively in g as required. ■

Exercise 8.2.4 *What are the correct relativizations of the previous two theorems?*

Exercise 8.2.5 *The above results can be extended by iterating the notions of "r.e. in" or more generally "Δ^0_2 in" as long as one includes the lower degrees. We say that A 1-REA if it is r.e. then we define n-REA by induction: A is $n+1$-REA if A is of the form $B \oplus W_e^B$ where B is n-REA. (REA stands for r.e. in and above.) Prove that any n-REA set A has an $f \equiv_T A$ such that any $g > f$ computes A. Do the same with Δ^0_2 replacing r.e. These results can be carried into the transfinite. Prove, for example, that $0^{(\omega)}$ has the same property.*

re1gen

Theorem 8.2.6 *If $A > 0$ is r.e. and $\mathcal{P}$ is a recursive notion of forcing then there is a 1-generic sequence $\langle p_s \rangle \leq_T A$ so that the corresponding 1-generic G is recursive in A as well.*

Proof. We build a 1-generic sequence p_s recursive in A. Let $f \leq_T A$ be the least modulus function for A. The requirements are

$$R_e : \text{for some } s,\ p_s \in S_e \text{ or } (\forall q \leq p_s)(q \notin S_e), \text{ where } S_e \text{ is } e\text{th } \Sigma_1 \text{ set of conditions.}$$

At stage s, we have a condition p_s. Note that we are thinking of P as a subset of $\mathbb{N}$ and so have the natural ordering $\leq$ on its members (and all of $\mathbb{N}$) as well as the forcing ordering $\leq_{\mathcal{P}}$. We say that R_e has been declared satisfied by stage s if there is a p_n with $n \leq s$ such that $p_n \in S_{e,f(s)}$. Find the least $e < s$ such that R_e has not yet been declared satisfied and such that $(\exists q \leq_{\mathcal{P}} p_s)(q \leq f(s)\ \&\ q \in S_{e,f(s)})$. For this e, choose the least such q and put $p_{s+1} = q$. If there is no such e, let $p_{s+1} = p_s$.

To verify that the construction succeeds, suppose for the sake of a contradiction that e_0 is least such that

$$\neg\exists s(p_s \in S_{e_0} \vee (\forall q \leq_{\mathcal{P}} p_s)(q \notin S_{e_0})).$$

Choose $s_0 > e_0$ such that $\forall i < e_0$ if there is a $p_s \in S_i$ then there is one with $s < s_0$ and $p_s \in S_{i,f(s_0)}$ (so by this stage we have already declared satisfied all higher priority requirements that are ever so declared). We claim that we can now recursively recover the entire construction and the values of $f(s)$ for $s \geq s_0$. As this would compute A recursively, we would have our desired contradiction. Consider what happens in the construction at each stage $s \geq s_0$ in turn. Suppose we have p_s. At stage s we look for the least $e < s$ such that $(\exists q \leq_{\mathcal{P}} p_s)(q \leq f(s)\ \&\ q \in S_{e,f(s)})$. There is no such $e < e_0$ by our choice of s_0. If e_0 itself were such an e, we would act for it and declare P_{e_0} to be satisfied, contrary to our choice of e_0. On the other hand, by our choice of e_0 there is a $q \leq_{\mathcal{P}} p_s$

with $q \in S_{e_0}$. We can find such a q recursively (because we know it exists). We did not find this q in the construction at stage s because either $q > f(s)$ or $q \in S_{e_0} - S_{e_0,f(s)}$. So we can now find a bound t on $f(s)$ by finding the stage at which q enters S_{e_0}. Given $t \geq f(s)$ we can calculate $f(s)$ as the least z such that $A_z \upharpoonright s = A_t \upharpoonright s$. Once we have $f(s)$ we can recursively determine what happened at stage s of the construction and in particular the value of p_{s+1}. Thus we can continue our recursive computation of $f(s)$ as claimed. ■

Relativizing Theorem re1gen 8.2.6 to C gives, for any C recursive notion of forcing $\mathcal{P}$, a $G \leq_T A$ which is C 1-generic for $\mathcal{P}$ for any $A >_T C$ which is r.e. in C.

Exercise 8.2.7 *The crucial property of the function f used in the above construction was that there is a uniformly recursive function computing $f(x)$ from any number greater than it. Prove that if there is a partial recursive $\varphi(x, s)$ such that $(\forall s \geq f(x))(\varphi(x, s) = f(x))$ then f is of r.e. degree.*

recohen **Corollary 8.2.8** *If $\mathbf{a} > \mathbf{0}$ is r.e. then there is Cohen 1-generic $G <_T A$ and so, for example, every countable partial order can be embedded in the degrees below $\mathbf{a}$.*

Similarly we have

Corollary 8.2.9 *If $\mathbf{a}$ is r.e. in $\mathbf{b}$ and strictly above it, then every partial lattice recursive in $\mathbf{b}$ can be embedded into $[\mathbf{b}, \mathbf{a})$.*

renomax **Corollary 8.2.10** *If $\mathbf{a}$ is r.e. then every maximal chain in $(\mathcal{D}(\leq \mathbf{a}), \leq_T)$ is infinite. In fact, there is no maximal element less than $\mathbf{a}$ in $(\mathcal{D}(\leq \mathbf{a}), \leq_T)$.*

Proof. Suppose $\mathbf{b} < \mathbf{a}$. Then $\mathbf{a}$ is r.e. in and strictly above $\mathbf{b}$. Relativizing Theorem re1gen 8.2.6 to a $B \in \mathbf{b}$ and using Cohen forcing gives us a $G \leq_T A$ which is Cohen 1-generic over B. So the degrees of $B \oplus G^{[i]}$ are in fact all between $\mathbf{b}$ and $\mathbf{a}$ and even independent. ■

We now apply Theorem re1gen 8.2.6 to provide the missing way of identifying the standard parts of effective successor models coded below $0'$ that we need to calculate the complexity of $Th(\mathcal{D}(\leq \mathbf{0}'))$.

sigma3ideal **Theorem 8.2.11** *If $A >_T C$, A is r.e. in C and I is an ideal in $\mathcal{D}(\leq \deg(C))$ such that $W = \{e : \deg(\Phi_e^C) \in I\} \in \Sigma_3^C$ then there is an exact pair G_0, G_1 for I below A.*

Proof. We provide a C-recursive notion of forcing $\mathcal{P}$ such that any 1-generic for $\mathcal{P}$ gives an exact pair for I and apply Theorem re1gen 8.2.6 relativized to C. The conditions of $\mathcal{P}$ are of the form $p = \langle p_0, p_1, F_p, n_p \rangle$ where $p_i \in 2^{<\omega}$, $|p_0| = |p_1| = |p|$, $F_p \in \mathbb{N}^{<\omega}$, $n_p \in \omega$ such that

$$(\forall i \in \{0,1\})(\forall \langle e, x, y \rangle)(\exists^{\leq 1} \langle w, m \rangle)\left(\langle e, x, y, w, m \rangle \in p_i\right).$$

We define V as expected $V(p) = p_0 \oplus p_1$. So for a 1-generic $\mathcal{G}$, we have $G_i = \cup\{p_i | p \in \mathcal{G}\}$. If $e \in W$, we want Φ_e^C to be coded into G_i. The unusual restriction above on conditions in P suggests how we intend to do this coding. Since $W \in \Sigma^C_3$ we have a relation $R \le_T C$ such that $e \in W \Leftrightarrow \exists x \forall y \exists z R(e,x,y,z)$. We denote the pairs of elements of W and their witnesses by $\hat{W} = \{\langle e,x\rangle : \forall y \exists z R(e,x,y,z)$. To calculate Φ^C_e for $e \in W$, our plan is to first choose an x such that $\langle e,x\rangle \in \hat{W}$. We then search for $\langle w,m\rangle$ such that $\langle e,x,y,w,m\rangle \in G_i$ and announce that $\Phi^C_e(y) = m$. The definition of P guarantees that this procedure gives at most one answer. The definition of the partial order $\le_{\mathcal{P}}$ below guarantees that this procedure makes only finitely many mistakes for any 1-generic. Genericity also guarantees that, when $\langle e,x\rangle \in \hat{W}$, it gives a total function.

The number n_p in our conditions acts as a bound for how far we have to search to sufficiently verify the Π_2 assertion that x is a witness that $e \in W$ (and so also that Φ^C_e is total). The set F_p tells us for which $\langle e,x\rangle$ we can make no further mistakes in our coding of Φ^C_e into $G^{\langle e,x\rangle}_i$ when we extend p. With this intuition, we define extension in $\mathcal{P}$ by $q \le_{\mathcal{P}} p$ iff

$$q_i \supseteq p_i, \qquad F_q \supseteq F_p, \qquad n_q \ge n_p,$$

and

$$\begin{aligned}(\forall i &\in \{0,1\})(\forall \langle e,x,y,w,m\rangle \in [|p|,|q|)(\langle e,x\rangle \in F_p \ \&\ \langle e,x,y,w,m\rangle \in q_i\\ &\to \Phi^C_{e,n_q}(y) = m \ \&\ \forall y' \le y \exists z \le n_q\,(R(e,x,y',z))\end{aligned}$$

Note that $\mathcal{P}$ is recursive in C.

Suppose that G_0, G_1 are given by a C-1-generic sequence $\langle p_s\rangle \le_T A$ as in Theorem re1gen 8.2.6 relativized to C. We claim that G_0, G_1 are an exact pair for I.

First assume that $\langle e,x\rangle \in \hat{W}$. We show that $\Phi^C_e \le_T G_i$. As the sets $\{p|\,\langle e,x\rangle \in F_p\}$ are obviously dense in $\mathcal{P}$, there is an s such that $\langle e,x\rangle \in F_{p_s}$. For any $\langle e,x,y,w,m\rangle \in p_t$ with $t > s$, $\Phi^C_e(y) = m$ by definition and so as noted above, the prescribed search procedure which is recursive in G_i returns only correct answers for $y > |p_s|$. Next, we claim that for each $y > |p_s|$, $i \in \{0,1\}$ and $m = \Phi^C_e(y)$ the Σ^C_1 sets $S_{e,x,y,m,i} = \{r|\exists w(\langle e,x,y,w,m\rangle \in r_i\}$ are dense below p_s. This guarantees that $\langle p_t\rangle$ meets each of these sets and so the search procedures are total and correctly compute $\Phi^C_e(x)$ for all but finitely many x. To see that these sets are dense below p_s, consider any $q \le p_s$ with no w such that $\langle e,x,y,w,m\rangle \in q_i$. Choose any $w > |q|$ and define an $r \le_{\mathcal{P}} q$ by making $|r| = \big\langle e,x,y,w,\Phi^C_e(y)\rangle + 1\big\rangle$, $r_i = q_i \cup \{\langle e,x,y,w,\Phi^C_e(y)\rangle\}$ (i.e. we let them be 0 at other points below the length), $F_r = F_q$ and letting n_r be the least $n \ge n_q$ such that $\forall y' \le y \exists z < n(R(e,x,y',z) \ \&\ \Phi^C_{e,n}(y)\downarrow)$ (one such exists since we are assuming that $\langle e,x\rangle \in \hat{W}$). Then $r {\le_{\mathcal{P}}} q$ and $r \in S_{e,x,y,m,i}$ as desired.

We next want to deal with the minimality conditions associated with the G_i being an exact pair for I. Suppose then that $\Phi^{G_0}_e = \Phi^{G_1}_e = D$ is total. We want to prove that $D \le \oplus\{\Phi^C_e : e \in F\}$ for some finite $F \subset W$. Consider the Σ_1 set S_e of conditions p:

$$S_e = \{p : \exists n\,(\Phi^{p_0}_e(n)\downarrow \ne \Phi^{p_1}_e(n))\downarrow\}.$$

By our assumption there is no $p_s \in S_e$ so we have a $p_s = p$ such that $\forall q \le_{\mathcal{P}} p(q \notin S_e)$. We claim that $D \le \oplus\{\Phi^C_e : \langle e,x\rangle \in F_p \cap \hat{W}\}$. For every $\langle e,x\rangle \in F_p \setminus \hat{W}$, let $y(e,x)$

be the least y such that $\neg\forall y' \le y\exists z R(e,x,y',z) \vee \Phi_e^C(y)\uparrow$. It is clear that there is no $q \le_{\mathcal{P}} p$ with any $\langle e,x,y,w,m\rangle \in q_i$ for $\langle e,x\rangle \in F_p \setminus \hat{W}$ and $y \ge y(e,x)$. Choose $q \le_{\mathcal{P}} p$ in $\langle p_s\rangle$ so that it has the maximal number of y's with some $\langle e,x,y,w,m\rangle \in q_i$ for $y < y(e,x)$ and $i \in \{0,1\}$. To compute $D(y)$ for $y > |q|$, we find a $t \in \mathcal{P}$ such that $t_i \supseteq q_i$, $\Phi_e^{t_0}(y)\downarrow = \Phi_e^{t_1}(y)\downarrow$, no elements not in q_i are added into t_i in columns $\langle e,x\rangle \in F_p \backslash \hat{W}$ and for any $\langle e,x,y,w,m\rangle \in t_i$ with $\langle e,x\rangle \in F_p \cap \hat{W}$, $\Phi_e^C(y) = m$. Such an extension exists because $\Phi_e^{G_0}(y)\downarrow = \Phi_e^{G_1}(y)\downarrow$ and by the maximality property of q and the definition of $\le_{\mathcal{P}}$, $G_i^{[\langle e,x\rangle]} = q_i^{[\langle e,x\rangle]}$ for $\langle e,x\rangle \in F_p \setminus \hat{W}$ and so there is such a $\hat{t} \in \langle p_s\rangle$. Finding one such t is clearly recursive in $\oplus\{\Phi_e^C : \langle e,x\rangle \in F_p \cap \hat{W}\}$. Thus we only need to show that any such t provides the right answer. If one such gave an answer different than that given by $\hat{t}$ (and so G_0 and G_1) then $\left\langle t_0, \hat{t}_1, F_p, n\right\rangle$ (where $n \ge n_q$ is large enough so that $\Phi_{e,n}^C(y)\downarrow$ for every $\langle e,x,y,w,m\rangle$ in t_0 or $\hat{t}_1$ with $\langle e,x\rangle \in F_p \cap \hat{W}$) would be an extension of p in S_e for the desired contradiction. ■

This Theorem completes the proof of Theorem Th(D<0') 7.3.5 that the theory of the degrees below $\mathbf{0}'$ is recursively isomorphic to true arithmetic. We can extend the result to all r.e. degrees.

Th<re **Exercise 8.2.12** *For every r.e.* $\mathbf{r} > \mathbf{0}$*,* $Th(\mathcal{D}(\le\mathbf{r}) \equiv_{1-1} Th(\mathbb{N})$.

??Explain??

Notes: Theorem redom 8.2.1 is due to Dekker [1954]; Theorem delta2dom 8.2.3 to Miller and Martin [1968]. We are not sure who first proved Corollary recohen 8.2.8 (presumably using a different method called r.e. permitting). The style of proof based directly on domination properties used here to prove Theorem relgen 8.2.6 is attributed to us in Soare [1987, Ch. VI Exercise 3.9] in the case of Cohen forcing. Theorem resigmaeideal **??** is in Shore [1981] which also is the original source of Exercise Th<re 8.2.12.

8.3 High and $\overline{GL_2}$ degrees

We now look at stronger domination properties and their relation to the jump classes $\mathbf{H}_1$ and $\bar{\mathbf{L}}_2$ below $\mathbf{0}'$ and their generalizations. Recall from §jumphier 4.6 that for $\mathbf{a} \le \mathbf{0}'$, $\mathbf{a} \in \mathbf{H}_1 \Leftrightarrow \mathbf{a}' = \mathbf{0}''$; $\mathbf{a} \in \mathbf{L}_2 \Leftrightarrow \mathbf{a}'' = \mathbf{0}''$. For degrees $\mathbf{a}$ not necessarily below $\mathbf{0}'$, $\mathbf{a} \in \mathbf{GL}_2 \Leftrightarrow (\mathbf{a} \vee \mathbf{0}')' = \mathbf{a}''$; $\mathbf{a} \in \mathbf{GH}_1 \Leftrightarrow \mathbf{a}' = (\mathbf{a} \vee \mathbf{0}')'$. It is also common to say that $\mathbf{a}$ is *high* if $\mathbf{a}' \ge \mathbf{0}''$. As it turns out these last are the degrees of dominant functions. Of course, $\mathbf{a} \in \overline{\mathbf{GL}}_2$ means that $\mathbf{a} \notin \mathbf{GL}_2$. We relativize these notions to degrees above $\mathbf{b}$ by writing, for example, $\mathbf{a} \in \overline{\mathbf{GL}}_2(\mathbf{b})$.

Let us begin by showing that there is there a dominant function. In fact, if $\mathcal{C}$ is any countable class of functions $\{f_i\}$ then there is function f which dominates all the f_i. For example, put $f(x) = \max\{f_i(x) : i < x\} + 1$. This construction requires a uniform list of all the functions f_i. For the recursive functions we know that $0''$ can compute such a list. Indeed, $Tot = \{e : \Phi_e \text{ total}\} \equiv_T 0''$ (Exercise Tot 4.5.4) and so there is a sequence f_i uniformly computable from $0''$ which then computes a dominant function as described. We can do better than this and avoid using totality. If $f(x) = \max\{\Phi_e(x) : e < x \ \&\ \Phi_e(x)\downarrow\}$ then $f \leq_T 0'$ and is also clearly dominant. We can even do a bit better and get away with functions of high degree.

martin **Theorem 8.3.1 (Martin's High Domination Theorem)** *A set A computes a dominant function f if and only if $0'' \leq_T A'$.*

Proof. Suppose first that $0'' \leq_T A'$. By the Shoenfield limit lemma (Theorem limitlemma 4.3.9) and the fact that $Tot \leq_T 0''$, there is an $h \leq_T A$ with $\lim_{s\to\infty} h(e,s) = Tot(e)$. We want to compute a function f recursively in A such that, for every e for which Φ_e is total, $f(x)$ is larger than $\Phi_e(x)$ for all but finitely many x. Any such f is dominant. To compute $f(x)$ we compute, for each $e < x$, both $\Phi_{e,t}(x)$ and $h(e,t)$ for $t \geq x$ until either the first one converges, say to y_e, or $h(e,t) = 0$. As, if Φ_e is not total, $\lim h(e,t) = 0$, one of these outcomes must happen. We set $f(x)$ to be one more than the maximum of all the y_e so computed for $e < x$. Note that $f \leq_T h \leq_T A$. It remains to verify that if Φ_e is total then $\Phi_e < f$. By our choice of h, $\exists s_0(\forall s \geq s_0)(h(e,s) = 1)$. So for $x > s_0$ when we calculate $f(x)$ we always find a t such that $\Phi_{e,t}(x)\downarrow = y_e$ and so $f(x) > \Phi_e(x)$ for all $x > s_0$.

For the other direction, suppose we have a dominant f. As Tot is Π^0_2 and computes $0''$, it suffices to show that it is also $\Sigma_2(f)$ as it would then be $\Delta_2(f)$ and so recursive in f'. We claim that

$$\forall x \exists s \Phi_{e,s}(x)\downarrow \ \Leftrightarrow\ \exists c \forall x \Phi_{e,f(x)+c}(x)\downarrow .$$

Suppose Φ_e is total (if not, then of course both conditions fail). Let $k(x) = \mu s \Phi_{k,s}(x)\downarrow$. Then k is recursive (because we know that$\forall x \Phi_e(x)\downarrow$). By hypothesis, f dominates k. Thus, the right hand side holds. This is a $\Sigma_2(f)$ formula as desired. ■

Now a look at the definitions shows that for $\mathbf{a} \leq_T \mathbf{0}'$, $\mathbf{a} \notin \mathbf{L}_2$ is equivalent to $\mathbf{0}'$ not being high relative to $\mathbf{a}$. Relativizing Theorem martin 8.3.1 to an $\mathbf{a} \leq_T \mathbf{0}'$ we see that $\mathbf{a} \notin \mathbf{L}_2$ if and only if no $f \leq_T 0'$ dominates every (total) function recursive in A. We can then handle $\overline{\mathbf{GL}}_2$ by relativizing to $\mathbf{a} \vee \mathbf{0}'$ to prove the following:

gl2 **Proposition 8.3.2** *A set $A \leq_T 0'$ has degree in $\overline{\mathbf{L}}_2$ if and only if $(\forall g \leq_T 0')(\exists f \leq_T A)(f \not< g)$. An arbitrary set A has degree in $\overline{\mathbf{GL}}_2$ if and only if $(\forall g \leq_T A \vee 0')(\exists f \leq_T A)(f \not< g)$.*

??Prove??

This says that, while sets that are not high do not compute dominant functions, if they are not too low they compute functions which are not dominated by any recursive function. This suffices for many applications.

gl21gen **Theorem 8.3.3** *If $A \notin GL_2$ then for any recursive notion of forcing $\mathcal{P}$ there is 1-generic sequence $\langle p_s\rangle \leq_T A$ and so the associated 1-generic G is also recursive in A.*

Proof. For any $g \leq_T A \vee 0'$, there is an $f \leq_T A$ not dominated by g. Without loss of generality we may take f to be strictly increasing. We first construct the function g that we want and then, using the associated f, we construct a 1-generic sequence p_s recursively in f (and so A). We again make use of the natural order $\leq$ on $P \subseteq \mathbb{N}$.

Let S_e list the Σ_1 subsets of P. As usual, we declare S_e to be satisfied at s if $(\exists n \leq s)(p_n \in S_{e,s})$. We define g by recursion using $0'$. Given $g(s)$, we want to determine $g(s+1)$. For each condition $p \leq g(s)+1$, ask $0'$ if $(\exists q \leq_{\mathcal{P}} p)(q \in S_e)$ for each $e \leq g(s)+1$. If such an extension exists, let x_e be the least x such that $(\exists q \leq_{\mathcal{P}} p)(q \leq x \;\&\; q \in S_{e,x})$. Put $g(s+1) = \max\{x_e | e \leq g(s)+1\}$.

We cannot use g itself in the construction of the desired 1-generic $\langle p_s \rangle$ because we want $\langle p_s \rangle \leq_T A$. But, since $g \leq_T A \vee 0'$, we can use an increasing $f \leq_T A$ not dominated by g. The construction of G is recursive in f (hence in A). At stage s, we have finite a condition p_s. For each $e \leq s$ not declared satisfied at s, see if $(\exists q \leq_{\mathcal{P}} p_s)(q < f(s+1) \;\&\; q \in S_{e,f(s+1)})$. If so, take the smallest such q for the least such e and let it be p_{s+1}. If not, $p_{s+1} = p_s$. The construction is recursive in f, hence in A. Thus $\langle p_s \rangle \leq_T A$ and the associated $G \leq_T A$ as well. Note that $p_s \leq f(s)$ by induction. Indeed $p_s \leq g(s)$ as well because $g(s)$ gives a bound on the witness required in the definition of p_s.

To verify that G is 1-generic suppose, for the sake of a contradiction, that there is a least e_0 such that

$$\neg \exists s(p_s \in S_{e_0} \vee (\forall p \leq_{\mathcal{P}} p_s)(p \notin S_{e_0})).$$

Choose s_0 such that, $(\forall i < e_0)[(\exists s)(S_i$ is declared satisfied at $s) \rightarrow S_i$ is declared satisfied by $s_0]$. Consider any $s > s_0$ at which $f(s+1) > g(s+1)$. By our choice of e_0, there is a $q \leq_{\mathcal{P}} p_s$ such that $q \in S_{e_0}$. Moreover, as $p_s \leq g(s)$, by definition of g there is one $\leq g(s+1)$ such that it belongs to $S_{e0,g(s+1)}$ as well. By our choice of s, $q \leq g(s+1) < f(s+1)$. Thus at stage $s+1$, we would act to extend p_s to a $p_{s+1} \in S_{e_0}$ for the desired contradiction. ■

cohenanr **Remark 8.3.4** *The function g we used in the above proof was actually recursive in $0'$. In fact, for Cohen forcing $g \leq_{wtt} 0'$. Thus we used the weaker property that for every function $g \leq_{wtt} 0'$ there is an $f \leq_T A$ not dominated by g. This property is called array non-recursiveness and is discussed in the next section.*

As for the r.e. degrees, having a 1-generic below a degree $\mathbf{a} \notin \mathbf{GL}_2$ provides a lot of information about the degrees below $\mathbf{a}$. For example, as in Corollary 8.2.8, we can embed every countable partial order below any $\mathbf{a} \notin \mathbf{GL}_2$. It is tempting to think that we could also prove the analog of Corollary 8.2.10 that every maximal chain in the degrees below $\mathbf{a}$ is infinite. This is true for $\mathbf{a} < \mathbf{0}'$ (Exercise 8.3.5) but was a long open question (Lerman [1983]). Cai [2012] has now proven that it is not true. There are $\mathbf{a} \notin \mathbf{GL}_2$ which are the tops of a maximal chain of length three.

max<0' **Exercise 8.3.5** *Prove that if $\mathbf{a} \leq \mathbf{0}'$ and $\mathbf{a} \notin \mathbf{L}_2$ then any maximal chain in the degrees below $\mathbf{a}$ is infinite.*

On the other hand, we can say quite a bit that is not true of arbitrary r.e. degrees about the degrees above $\mathbf{a}$ when $\mathbf{a} \notin \mathbf{GL}_2$.

Definition 8.3.6 *A degree* $\mathbf{a}$ *has the* cupping property *if* $(\forall \mathbf{c} > \mathbf{a})(\exists \mathbf{b} < \mathbf{c})(\mathbf{a} \vee \mathbf{b} = \mathbf{c})$.

gl2cup **Theorem 8.3.7** *If* $\mathbf{a} \in \overline{\mathbf{GL}}_2$ *then* $\mathbf{a}$ *has the cupping property. Indeed, if* $A \notin GL_2$ *and* $C >_T A$ *then there is* $G \not\geq_T A$ *such that* $A \oplus G \equiv_T C$ *and* G *is Cohen* 1*-generic.*

Proof. We need to add requirements $R_e : \Phi_e^G \neq A$ to the proof of Theorem gl21gen 8.3.3 for Cohen forcing (making all the requirements into a single list Q_e) and code C into G as well (so as to be recoverable from $A \oplus G$). In the definition of $g(s+1)$ in that proof, for each $p \leq g(s)+1$ look as well for $q_0, q_1 \supseteq p$ and x such that $q_0|_e q_1$. Then make $g(s+1)$ also bound the least such extensions τ_0, τ_1 for each $e, p \leq g(s)+1$ for which such extensions exist.

Again choose $f \leq_T A$ strictly increasing and not dominated by g. The construction is done recursively in $f \oplus C$. At stage s we have p_s and we look for the least e such that Q_e has not yet been declared satisfied and for which there is either a $q \leq_{\mathcal{P}} p$ with $q \leq f(s+1)$ that would satisfy Q_e as before if it is an S_i or a pair of strings $q_0, q_1 \supseteq p_s$ with $q_i \leq f(s+1)$ such that $q_0|_e q_1$ if $Q_e = R_i$. Let e be the least for which there are such extensions. If $Q_e = S_i$ choose q as before. If it is R_i Let q be the q_j such that $\Phi_e^{q_j}(x) \downarrow \neq A(x)$. We then let $p_{s+1} = q\hat{\ }C(s)$ and declare Q_e to be satisfied. If there is no such e, we let $p_{s+1} = p_s\hat{\ }C(s)$. Note that $p_{s+1} \leq f(s+1)+1$ (the extra 1 comes from appending $C(s)$).

Since the construction is recursive in $f \oplus C$ and $f \leq_T A \leq_T C$, we have $G \leq_T C$. But, $C \leq_T \langle p_s \rangle$ because $C(s) = p_{s+1}(|p_{s+1}|)$. However, $\langle p_s \rangle \leq_T A \vee G$ because $f \leq_T A$ tells how to compute each stage from the given p_s to the choice of q. Then G tells us the last extra bit at the end of p_{s+1}.

To verify that G has the other required properties suppose e_0 is least such that Q_e fails. Assume that by stage s_0 we have declared all requirements with $e' < e_0$ which will ever be declared satisfied to be satisfied. Consider a stage $s > s_0$ at which $f(s+1) > g(s+1)$. If $Q_e = S_i$ then we argue as in the previous theorem. If $Q_e = R_i$ and there were any $q_0, q_1 \supseteq p_s$ with $q_0|_e q_1$ then would have taken one of them as our q and declared $Q_e = R_i$ to be satisfied contrary to our choice of e_0. On the other hand, if there are no such extensions, then as usual Φ_e^G is recursive if total and so R_i would also succeed contrary to our assumption. ■

renoncup **Remark 8.3.8** *Not every r.e. degree has the cupping property.*

For other results about $\overline{\mathbf{GL}}_2$ degrees it is often useful to strengthen Theorem gl21gen 8.3.3 to deal with notions of forcing recursive in A rather than just recursive ones.

gl2genseq **Theorem 8.3.9** *For* $A \in \overline{GL}_2$*, given an* A *recursive notion of forcing* $\mathcal{P}$ *and a sequence* D_n *of dense sets uniformly recursive in* $A \vee 0'$ *(or with a density function* $d(n,p) \leq_T A \vee 0'$*) there is a generic sequence* $\langle p_s \rangle \leq_T A$ *meeting all the* D_n*. Of course, the generic* G *associated with the sequence is recursive in* A *as well.*

Proof. Let m_K be the least modulus function for $K = 0'$ and let $\Psi_n^{A\oplus K} = D_n$, i.e. the Ψ_n uniformly compute membership in D_n. We define $g \le_T A \vee 0'$ by recursion. Given $g(s)$ we find, for each $p, n \le g(s)+1$ the least q such that $q \le_P p$ and $q \in D_n$ as witnessed by a computations of $\Psi_{n,u}^{A\oplus K_u \restriction u}(n) = 1$ where K_u is the same as K on the use from K in this computation. Next we let $g(s+1)$ be the least number larger than q, u and $m_K(u)$ for all of these q and u as well as $m_K(g(s)+1)$. As $g \le_T A \vee 0'$ and $A \in \overline{GL}_2$ there is an increasing $f \le_T A$ not dominated by g.

We construct the sequence $\langle p_s \rangle$ recursively in $f \le_T A$. At stage s we have p_s. Our plan is to satisfy the requirement of meeting D_n for the least n for which we do not seem to have done so yet and for which we can find an appropriate extension of p_s when we restrict our search to $q \le f(s+1)$ as well as our use of $0'$ to what we have at stage $f(s+1)$. More formally, we determine (recursively in A) for which D_n $(n \le s)$ there is a $t \le s$ such that $\Psi_n^{(A\oplus K_{f(s+1)})\restriction f(s+1)}(p_t) = 1$. Among the other $n \le s$, we search (again recursively in A) for one such that $(\exists q \le_{\mathcal{P}} p_s)(q \le f(s+1)\ \&\ \Psi_n^{(A\oplus K_{f(s+1)})\restriction f(s+1)}(q) = 1)$. If there is one we act for the least such n by letting p_{s+1} be the least such q for this n. If not, let $p_{s+1} = p_s$. Note that $p_{s+1} \le f(s+1)$ by the restriction on the search space and $p_{s+1} \le g(s+1)$ as well since $g(s+1)$ also bounds the least witness by the definition of g.

We now claim that for each n there is a $p_s \in D_n$. If not, suppose, for the sake of a contradiction, that n is the least counterexample. Choose s_0 such that for all $m < n$ there is $t < s_0$ such that $p_t \in D_m$ and indeed such that $\Psi_m^{(A\oplus K_{s_0})\restriction s_0}(p_t) = 1$ and $K_{s_0} \restriction u = K \restriction u$ where u is the use of this computation of Ψ_m at p_t. Thus, by construction, we never act for $m < n$ after s_0. As g does not dominate f we may choose an $s > s_0$ with $f(s+1) > g(s+1)$. At stage s we have p_s and $p_t \notin D_n$ for all $t \le s$ in the sense required, i.e. $\Psi_n^{(A\oplus K_{f(s+1)})\restriction f(s+1)}(p_t) = 0$ since any computation of this form gives the correct answer by our definition of $g(s+1)$ and the fact that $f(s+1) > g(s+1)$. There is a $q \le_{\mathcal{P}} p_s$ with $q \in D_n$ and the least such is less than $f(s+1)$ and $\Psi_n^{(A\oplus K_{f(s+1)})\restriction f(s+1)}(q) = 1$ with the computation being a correct one from $A \oplus K$ by the definition of $g(s+1) < f(s+1)$. Thus we would take the least such q to be $p_{s+1} \in D_n$ for the desired contradiction. ■

We now give a couple of applications that play a crucial role in our global analysis of definability in $\mathcal{D}(\le \mathbf{0}')$. ??Later also for $\mathcal{D}$ and, in particular, of the jump operator ??. The first is a jump inversion theorem that ??strengthens and (check original)?? generalizes Shoenfield's.

ompleteness

Theorem 8.3.10 ($\overline{\mathbf{GL}}_2$ jump inversion) *If $A \in \overline{GL}_2$, $C \ge_T A \vee 0'$, and C is r.e. in A, then there is a $B \le_T A$ such that $B' \equiv_T C$.*

Proof. Let C_s be an enumeration of C recursive in A. We want a notion forcing recursive in A and a collection of dense sets D_n such that for any $\langle D_n \rangle$ generic G, $G' \equiv_T C$. This time, our notion of forcing has conditions $p \in 2^{<\omega}$. The definition of extension for $\mathcal{P}$ is a bit tricky. If $q \supseteq p$ and

$$\langle e, x \rangle \in [|p|, |q|) \Rightarrow [C_{|p|}(x) = q(\langle e, x \rangle) \text{ or } \exists n \le e\,(\Phi_n^p(n) \uparrow\ \&\ \Phi_n^q(n) \downarrow)]$$

we say that $q \leq_1 p$. Now this relation is clearly recursive in A since A computes $C_{|p|}$ for each p. However, it need not be transitive (Exercise). We let $\leq_{\mathcal{P}}$ be its transitive closure. As, given any $r \supseteq p$, there are only finitely many q's with $r \supseteq q \supseteq p$ we can check all possible routes via $\leq_1$ from p to r recursively in A and so $\leq_{\mathcal{P}}$ is also recursive in A. The plan for coding C into G' uses the Shoenfield limit lemma and partially explains the notion of extension. It guarantees that $e \in C \Rightarrow G^{[e]} =^* \omega$ while $e \notin C \Rightarrow G^{[e]} =^* \emptyset$. Thus $e \in C \Leftrightarrow \lim_s G(\langle e, s\rangle = 1$ and so $C \leq_T G'$. Suppose we have a generic sequence $\langle p_s \rangle \leq_T A$ for some collection of dense sets as in Theorem gl2genseq 8.3.9. The definition of extension guarantees that coding mistakes can happen in column e only when $\Phi_n^{p_s}(n)$ first converges for some $n \leq e$. Thus $C \leq_T G'$.

Our first class of dense sets include the trivial requirements and in addition force the jump of G in the hope of making $G' \leq_T C$:

$$\begin{array}{rcl} D_{m,j} & = & \{p : |p| \geq j \ \& \ [\Phi_m^p(m) \downarrow \text{ or } (\forall q \supseteq p)(\Phi_m^q(m) \uparrow \\ \text{or } [(\exists e & < & m)(\exists \langle e, x\rangle \in [|p|, |q|)(C_{|p|}(e) \neq q(\langle e, x\rangle) \text{ but } \neg(\exists n \leq e)(\Phi_n^p(n) \uparrow \ \& \ \Phi_n^q(n) \downarrow)])\} \end{array}$$

Note that, after we use A to compute $C_{|p|}$, membership in $D_{m,j}$ is a Π_1 property and so recursive in $0'$. Thus, the $D_{m,j}$ are uniformly recursive in $A \vee 0'$. We must argue that they are dense. Consider any p. We can clearly extend it to a q with $|q| \geq j$ by making $q(\langle e, x\rangle) = C_{|p|}(e)$ for $\langle e, x\rangle \in [|p|, j)$. So we may as well assume that $|p| \geq j$. If $\Phi_m^p(m) \downarrow$ then $p \in D_{m,j}$ and we are done. So suppose $\Phi_m^p(m) \uparrow$. If there is $q \supseteq p$ such that $\Phi_m^q(m) \downarrow$ and $(\forall e < m)(\forall \langle e, x\rangle \in [|p|, |q|)[C_{|p|}(x) = q(\langle e, x\rangle)$ or $\exists n \leq e \, (\Phi_n^p(n) \uparrow \ \& \ \Phi_n^q(n) \downarrow)]$, $q \leq_{\mathcal{P}} p$ by definition (because $\Phi_m^p(m) \uparrow$ while $\Phi_m^q(m) \downarrow$ so any violation of coding is allowed for $e \geq m$) and is in $D_{m,j}$. If there is no such q then $p \in D_{m,j}$ by definition.

Now we verify that $G = \cup p_s$ has the desired properties. By Theorem gl2genseq 8.3.9, $G \leq_T A$. To see that $C \leq_T G'$ consider any e. Let s be such that $(\forall i \leq e)(\Phi_i^G(i) \downarrow \Rightarrow \Phi_i^{p_s}(i) \downarrow \ \& \ i \in C \Rightarrow i \in C_{|p_s|})$. It is clear from the definition of $\leq_{\mathcal{P}}$ that for any $t > s$ and $\langle i, x\rangle \in [|p_s|, |p_t|)$ with $i \leq e$, $\langle i, x\rangle \in p_t \Leftrightarrow i \in C$. Thus $C(e) = \lim_t G(\langle e, t\rangle$ and so $C \leq_T G'$ by the Shoenfield limit lemma. For the other direction we want to compute $G'(e)$ recursively in C. (Of course, $A \leq_T C$ and so then is $\langle p_s \rangle$.) Suppose we have, by induction, computed an s as above for $e - 1$. We can now ask if $e \in C$. If so, we find a $u \geq t \geq s$ such that $e \in C_{|p_t|}$ and $p_u \in D_{e,|p_t|}$. If $\Phi_e^{p_u}(e) \downarrow$, then, of course, $e \in G'$. If $\Phi_e^{p_u}(e) \uparrow$ but $e \in G'$, then there would be a $v > u$ such that $\Phi_e^{p_v}(e) \downarrow$ and, of course, $p_v \leq_{\mathcal{P}} p_u$. This would contradict the fact that $p_u \in D_{e,|p_t|}$ by our choice of s and t and the definitions of $D_{e,|p_t|}$ and $\leq_{\mathcal{P}}$. ■

Shj2 **Corollary 8.3.11 (Shoenfield Jump Inversion Theorem)** *For all $C \geq 0'$ there is $B < 0'$ such that $B' \equiv_T C$ if and only if C is r.e. in $0'$.*

Proof. The "only if" direction is immediate. The "if" direction follows directly from the Theorem by taking $A = 0'$. ■

For later applications we now strengthen the above jump inversion theorem to make $B <_T A$.

ompleteness **Theorem 8.3.12** *If $A \in \overline{GL}_2$, $C \geq_T A \vee 0'$, and C is r.e. in A, then there is $B <_T A$ such that $B' \equiv_T C$.*

Proof. In addition to the requirements of Theorem gl2completeness 8.3.10, we need to make sure that $\Phi_i^G \neq A$ for each i. To do this we modify the definition of extension to also allow violations of the coding requirements for e when we newly satisfy one of these diagonalization requirements for $i \leq e$. (As we did above for making $\Phi_i^G(i)\downarrow$.) We say $q \leq_1 p$ if

$$\langle e,x\rangle \in [|p|,|q|) \Rightarrow [C_{|p|}(x) = q(\langle e,x\rangle) \text{ or}$$
$$\exists n \leq e\,([\Phi_n^p(n)\uparrow\ \&\ \Phi_n^q(n)\downarrow] \text{ or } [\exists y \Phi_n^q(y)\downarrow\neq A(y)\ \&\ \neg\exists y\Phi_n^p(y)\downarrow\neq A(y)])\,.$$

Again $\leq_{\mathcal{P}}$ is defined as the transitive closure of this relation and it is recursive in $A \vee 0'$ as before. We then adjust the $D_{m,j}$ accordingly

$$\begin{aligned} D_{m,j} &= \{p : |p| > j\ \&\ [\Phi_m^p(m)\downarrow \text{ or } (\forall q \supseteq p)(\Phi_m^q(m)\uparrow \\ \text{or } [(\exists e &< m)(\exists\langle e,x\rangle \in [|p|,|q|)(C_{|p|}(e) \neq q(\langle e,x\rangle) \text{ but} \\ \neg(\exists n &\leq e)([\Phi_n^p(n)\uparrow\ \&\ \Phi_n^q(n)\downarrow]\ \&\ \neg(\exists y)[\Phi_n^q(y)\downarrow\neq A(y)\ \&\ \neg\exists y\Phi_n^p(y)\downarrow\neq A(y)])]\}. \end{aligned}$$

We also need dense sets that guarantee that $\Phi_e^G \neq A$:

$$\begin{aligned} D_i &= \{p|(\exists x)(\Phi_i^p(x)\downarrow\neq A(x) \text{ or} \\ (\forall q_0, q_1 &\supseteq p)(\forall x < |q_0|,|q_1|)[\neg(\Phi_i^{q_0}(x)\downarrow\neq \Phi_i^{q_1}(x)\downarrow) \text{ or} \\ ((\exists e &< i)(\exists\langle e,x\rangle \in [|p|,|q|)(\exists j \in \{0,1\})[(C_{|p|}(e) \neq q_i(\langle e,x\rangle) \text{ but} \\ \neg(\exists n &\leq i)([\Phi_n^p(n)\uparrow\ \&\ \Phi_n^q(n)\downarrow]\ \&\ \neg(\exists y)[\Phi_n^q(y)\downarrow\neq A(y)\ \&\ \neg\exists y\Phi_n^p(y)\downarrow\neq A(y)])]\}. \end{aligned}$$

The proof now proceeds as in the previous Theorem. The arguments for all the verifications are now essentially the same as there and are left as an exercise.?? ■

Exercise 8.3.13 *Verify that the notion of forcing and classes of dense sets specified in the proof of Theorem stgl2completeness 8.3.12 suffice to actually prove it.*

Exercise 8.3.14 *Prove that if A is r.e. and $C \geq_T 0'$ is r.e. in A then there is a $B \leq_T A$ such that $B' \equiv_T C$. Indeed we may also make $B <_T A$. ??Hint:??*

The next result says that every $\mathbf{a} \in \overline{\mathbf{GL}}_2$ is **RRE** (*relatively recursively enumerable*), i.e. there is a $\mathbf{b} < \mathbf{a}$ such that $\mathbf{a}$ is r.e. in $\mathbf{b}$ and a bit more.

gl2rre **Theorem 8.3.15** *If $\mathbf{a} \in \overline{\mathbf{GL}}_2$ then there is $\mathbf{b} < \mathbf{a}$ such that $\mathbf{a}$ is r.e. in $\mathbf{b}$ and $\mathbf{a}$ is in $\overline{\mathbf{GL}}_2(\mathbf{b})$, i.e. $(\mathbf{a} \vee \mathbf{b}')' < \mathbf{a}''$.*

Proof. Let $\mathbf{a} \in \overline{\mathbf{GL}}_2$. We'll use a notion of forcing $\mathcal{P}$ with conditions $p = \langle p_0, p_1, p_2\rangle$, $p_i \in 2^{<\omega}$ such that

1. $|p_0| = |p_1|$, $p_0(d_n) = A(n)$, $p_1(d_n) = 1 - A(n)$ where d_n is nth place where p_0, p_1 differ and
2. $(\forall e < |p_0 + p_1|)(e \in p_0 \oplus p_1 \Leftrightarrow \exists x(\langle e, x\rangle \in p_2))$.

As expected, our generic set $G_0 \oplus G_1 \oplus G_2$ is given by $V(p) = p_0 \oplus p_1 \oplus p_2$. The idea here is that if we can force p_0, p_1 to differ at infinitely many places while still making our generic sequence recursive in A, the first clause in the definition of $\leq_{\mathcal{P}}$ guarantees that $G_0 \oplus G_1 \equiv_T A$. The second clause works towards making $G_0 \oplus G_1$ r.e. in G_2 with the intention being that $\deg(G_2) = \mathbf{g}_2$ is to be the $\mathbf{b}$ required by the theorem. Extension in the notion of forcing is defined in the simplest way as $q \leq_{\mathcal{P}} p \Leftrightarrow q_i \supseteq p_i$ but note that this only applies to p and q in $\mathcal{P}$ and not all q with $q_i \supseteq p_i$ are in $\mathcal{P}$ even if $p \in \mathcal{P}$. The notion of forcing is clearly recursive in A.

We now define the dense sets needed to satisfy the requirements of the Theorem. We begin with $D_{2n} = \{p : p_0, p_1 \text{ differ at at least } n \text{ points}\}$. These sets are clearly recursive in A. We argue that these are dense by induction on n. Suppose D_{2n} is dense. To show that D_{2n+2} is dense, it suffices, for any given $p \in D_{2n} - D_{2n+2}$, to find a $q \leq_{\mathcal{P}} p$ in D_{2n+2}. Let $q_0 = p_0\hat{}A(n)$, $q_1 = p_1\hat{}(1 - A(n))$. Choose $i \in \{0, 1\}$ such that $q_i(|p_0|) = 1$. Define $q_2 \supseteq p_2$ by choosing x large and setting $q_2(\langle 2|p_0| + i, x\rangle) = 1$ and $q_2(z) = 0$ for all $z \notin \mathrm{dom}(p_2)$ and less than $\langle 2|p_0| + i, x\rangle$. Now $q = \langle q_0, q_1, q_2\rangle$ satisfies the requirements to be a condition in P. It obviously extends p and is in D_{2n+2}.

For any generic recursive in A which meets all the D_{2n}, $G_0 \oplus G_1 \equiv_T A$ and $G_0 \oplus G_1$ is r.e. in G_2.

We also want dense sets similar in flavor to those of the previous theorems to force the jump of G_2 to make $(\mathbf{a} \vee \mathbf{g}_2')' < \mathbf{a}''$. Let

$$\begin{aligned} D_{2n+1} \;\; = \;\; & \{p : \Phi_n^{p_2}(n)\downarrow \text{ or } (\forall \sigma \supseteq p_2) \\ & (\Phi_n^{\sigma}(n)\uparrow \text{ or } (\exists \langle e, x\rangle \in \sigma)((p_0 \oplus p_1)(e) = 0)\}. \end{aligned}$$

For $p \in P$, membership in D_{2n+1} is a $0'$ question and so these sets are recursive in $A \vee 0'$. We want to prove that they are dense. Suppose have a $p \in P$ and so we want a $q \leq_{\mathcal{P}} p$ with $q \in D_{2n+1}$. We may suppose that $\Phi_n^{p_2}(n)\uparrow$ and that the second clause fails for p as otherwise we would already be done. Thus we have a $\sigma \supseteq p_2$ such that $\Phi_n^{\sigma}(n)\downarrow$ but $\neg(\exists\langle e, x\rangle \in \sigma)((p_0 \oplus p_1)(e) = 0)$. We claim that there is a $q \leq_{\mathcal{P}} p$ such that $q_2 \supseteq \sigma$ and so $\Phi_n^{q_2}(n)\downarrow$ and $q \in D_{2n+1}$ as required. The only issue is that there may be some $\langle j, y\rangle \in \sigma$ with $j > |p_0 \oplus p_1|$. If so, we must define q_0 and q_1 accordingly, i.e. $j \in q_0 \oplus q_1$. So if j is even, we want $\frac{j}{2} \in q_0$; if it is odd, $\frac{j-1}{2} \in q_1$. We now define q_0, q_1 at the appropriate element ($\frac{j}{2}$ or $\frac{j-1}{2}$) to both be 1. Elsewhere we let both q_0 and q_1 be 0. Thus we have not added any points at which q_0 and q_1 differ beyond those in p_0, p_1). Now we extend σ to q_2 by adding $\langle e, y\rangle$ for some large y if $(q_0 \oplus q_1)(e) = 1$ and $e \geq |p_0 \oplus p_1|$ and wherever not yet defined we let $q_2(z) = 0$. Thus $q \in P$ and is the desired extension of p in D_{2n+1} as $\Phi_n^{q_2}(n) = \Phi_n^{\sigma}(n)\downarrow$.

We now let $\langle p_s\rangle \leq_T A$ be a generic sequence meeting every D_n as given by Theorem gl2genseq 8.3.9. We have already seen that $G_0 \oplus G_1 \equiv_T A$ and it is r.e. in $G_2 \leq_T A$. If we can show that $(A \oplus G_2')' <_T A''$ then we will be done as this clearly implies that $G_2 <_T A$. We first claim that $G_2' \le_T A \vee 0'$. To see if $n \in G_2'$, recursively in $A \vee 0'$ find an s such that $p_s \in D_{2n+1}$. Then we claim that $n \in G_2' \Leftrightarrow \Phi_n^{p_{s,2}}(n)\downarrow$. If $\Phi_n^{p_2}(n)\downarrow$, then we are done. If not, then $(\forall \sigma \supseteq p_{s,2})\,(\Phi_n^{\sigma}(n)\uparrow$ or $(\exists \langle e, x\rangle \in \sigma)((p_0 \oplus p_1)(e) = 0))$ and by definition of membership and extension in $\mathcal{P}$, $\Phi_n^{p_{t,2}}(n)\uparrow$ for every $p_{t,2}$ for $t \ge s$. Thus $\Phi_n^{G_2}(n)\uparrow$ as desired. As $G_2' \le_T A \vee 0'$, $(A \oplus G_2') = A \vee 0'$ and so as $A \notin GL_2$, $(A \oplus G_2')' = (A \vee 0')' <_T A''$ as required. ■

Exercise 8.3.16 *If $A >_T 0$ is r.e. and $C \ge_T 0'$ is r.e. in A then there is a $B \le_T A$ such that $B' \equiv_T C$. Indeed we may also make $B <_T A$. Hint: Build β_s finite extensions that obey a coding rule for columns for $e \le c(s) \le s$ (so that we can enumerate C recursively in A) except that we can violate this rule so as to force jump as above; search below $m_A(s+1)$ for extensions forcing the jump for $e \le s$ that obey rule. Also search for extensions with Φ_e giving different answers and allow violations in columns $> e$ when we satisfy this requirement by choosing one that gives an answer other than A.*

We can now deduce a result that plays a major role in our analysis of definability in $\mathcal{D}(\le \mathbf{0}')$. ??definition of the Turing jump in $\mathcal{D}$ and many related results.??

sigma3exact **Theorem 8.3.17** *If $\mathbf{b} <_T \mathbf{a}$ and $\mathbf{a} \in \overline{GL}_2(\mathbf{b})$ and $\mathcal{I}$ is a Σ_3^B ideal in $\mathcal{D}(\le \mathbf{b})$ then there is an exact pair for $\mathcal{I}$ below $\mathbf{a}$.*

Proof. By Theorem gl2rre 8.3.15 (relativized to $\mathbf{b}$) there is a $\mathbf{c}$ such that $\mathbf{b} \le \mathbf{c} < \mathbf{a}$ and $\mathbf{a}$ is r.e. in $\mathbf{c}$. So $\mathcal{I}$ is also Σ_3^C. Now, by Theorem resigma3ideal 8.2.11, we have the desired exact pair. ■

2codesigma3 **Theorem 8.3.18** *If $A \in \mathbf{a} \in \overline{\mathbf{GL}}_2$ and $S \in \Sigma_3^A$ then there is an embedding of a nice effective successor model (with the appropriate partial lattice structure) in the degrees below $\deg(A)$ and an exact pair $\mathbf{x}, \mathbf{y} \le \mathbf{a}$ for the ideal generated by the $\mathbf{d}_n$ with $n \in S$. (Remember that the $\mathbf{d}_n$ are the degrees representing $n \in \mathbb{N}$ in the effective successor model.*

Proof. Given $A \in \overline{GL}_2$ and $S \in \Sigma_3^A$, Theorem gl2rre 8.3.15 gives us a $B < A$ such that A is r.e. in B and A is $\overline{GL}_2(B)$. Since $A' \ge A \vee 0'$ and is r.e. in it, Theorem gl2completeness 8.3.10 relativized to B gives us a $\hat{B} < A$ (with $B \le_T \hat{B}$) such that $\hat{B}' \equiv A'$ and so $\Sigma_3^{\hat{B}} = \Sigma_3^A$, Moreover, A is r.e. in $\hat{B}$ because it was r.e. in $B \le_T \hat{B}$. The result now follows by using Theorem re1gen 8.2.6 and Exercise latemb1gen ?? to embed an effective successor model between $\hat{B}$ and A and then Theorem resigma3ideal 8.2.11 to pick out the ideal generated by the associated $\mathbf{d}_n$ for $n \in S$ as the set $\{e | \exists n(\Phi_e^{\hat{B}} \in \mathbf{d}_n)\}$ is itself $\Sigma_3^{\hat{B}} = \Sigma_3^A$ as is then $\{e | (\exists n \in S)(\Phi_e^A \in \mathbf{d}_n)\}$. ■

??Simplify second or as corollary to first??Below a $\mathbf{H}_1$ or $\mathbf{GH}_1$ degree?? Minimal degree in ?? others here??complementation??

gl2below **Exercise 8.3.19** *Prove that every degree has a $\mathbf{GL}_2$ degree below it.*

Exercise 8.3.20 *Prove that every recursive lattice $\mathcal{L}$ with* 0 *and* 1 *can be embedded in* $\mathcal{D}(\leq\mathbf{a})$ *preserving* 0 *and* 1 *for any* $\mathbf{a} \in \overline{\mathbf{GL_2}}$.

Notes: Theorem 8.3.1 is due to Martin [1966]. Its very useful consequence, Proposition 8.3.2 is from Jockusch and Posner [1978] which also contains a version of Theorem 8.3.3 for Cohen forcing, Exercises 8.3.5 and 8.3.19 as well as Theorem 8.3.10. The version given here of Theorem 8.3.3 and the more general Theorem 8.3.9 as well as Theorem 8.3.15 come from Cai and Shore [2012]. Corollary 8.3.11 was originally proved in Shoenfield [1959]. The original direct proof of (a stronger version of) Theorem 8.3.18 is in Shore [2007]. Remark 8.3.8 follows, for example, from Slaman and Steel [1989, Theorem 3.1] or Cooper [1989]. Theorem 8.3.7 is from Jockusch and Posner [1978].

def<0'

8.4 Definability and Biinterpretability in $\mathcal{D}(\leq \mathbf{0}')$

We already know that the theory of $\mathcal{D}(\leq\mathbf{0}')$ is (recursively) equivalent to true first order arithmetic and so as complicated as possible. We now want attack the problem of determining which subsets of, and relations on, $\mathcal{D}(\leq\mathbf{0}')$ are definable in the structure. The interpretation of $\mathcal{D}(\leq\mathbf{0}')$ in $\mathbb{N}$ gives a necessary condition. Only subsets and relations definable in arithmetic can possibly be definable in $\mathcal{D}(\leq\mathbf{0}')$. Our goal is to prove that, if they are also invariant under the double jump, then the are, in fact, definable in $\mathcal{D}(\leq\mathbf{0}')$.

Definition 8.4.1 *A relation $R(x_1, \ldots, x_n)$ on degrees is invariant under the double jump if, for all degrees $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and $\mathbf{y}_1, \ldots, \mathbf{y}_n$ such that $\mathbf{x}_i'' = \mathbf{y}_i''$ for all $i \leq n$, $R(\mathbf{x}_1, \ldots, \mathbf{x}_n) \Leftrightarrow R(\mathbf{y}_1, \ldots, \mathbf{y}_n)$.*

We begin with the subsets of $\mathcal{D}(\leq \mathbf{0}')$ and, in particular, with the basic question of definably determining the double jump of a degree $\mathbf{a} \leq \mathbf{0}'$. (This would actually suffice to show that all subsets of $\mathcal{D}(\leq\mathbf{0}')$ invariant under double jump and definable in arithmetic are definable in $\mathcal{D}(\leq\mathbf{0}')$ but as we prove more later we omit this argument.) The crucial point is that the sets we can code below an r.e. or $\overline{\mathbf{GL}}_2$ degree $\mathbf{a}$ are precisely the ones Σ_3^A. We use this to determine $\mathbf{a}''$ via the following characterization of the double jump.

sigma3=dj **Proposition 8.4.2** *For any sets A and B, $A'' \equiv_T B''$ if and only if $\Sigma_3^A = \Sigma_3^B$. Indeed, for any $n \geq 1$, $A^{(n)} \equiv_T B^{(n)}$ if and only if $\Sigma_{n+1}^A = \Sigma_{n+1}^B$.*

Proof. The hierarchy theorem 4.5.1 says that, for any set X and $n \geq 1$, $\Sigma_{n+1}^X = \Sigma_1^{X^{(n)}}$. On the other hand, for any Z and W, $\Sigma_1^Z = \Sigma_1^W$ iff $Z \equiv_T W$ since the equality implies that both Z and $\bar{Z}$ (W and $\bar{W}$) are Σ_1, i.e. r.e., in W (Z) and so each is recursive in the other. Thus if $\Sigma_{n+1}^A = \Sigma_{n+1}^B$ then $\Sigma_1^{A^{(n)}} = \Sigma_1^{B^{(n)}}$ and so $A^{(n)} \equiv_T B^{(n)}$ as required. ■

defL2 **Theorem 8.4.3** *The set $\mathbf{L}_2 = \{\mathbf{x} \leq \mathbf{0}' | \mathbf{x}'' = \mathbf{0}''\}$ is definable in $\mathcal{D}(\leq\mathbf{0}')$.*

Proof. Our analysis of coding in models of arithmetic in Proposition 6.4.9 and preceding Theorem 7.3.5 (which is really part of the proof of that theorem), shows that we have a way to, definably in $\mathcal{D}(\leq\mathbf{0}')$, pick out, via correctness conditions, parameters $\bar{\mathbf{p}}$ that define structures $\mathcal{M}(\bar{\mathbf{p}})$ isomorphic to $\mathbb{N}$. (The crucial point here is Theorem 8.2.11 which says that there is an exact pair for the $\Sigma_3^{\bar{\mathbf{p}}_0}$ ideal generated by the standard part of the model below $\mathbf{0}'$ as it is r.e. in and strictly above $\bar{\mathbf{p}}_0$.) Also note that, by Proposition 6.4.9, any set S coded in $\mathcal{M}(\bar{\mathbf{p}})$ them by a pair $\mathbf{g}_0, \mathbf{g}_1$ and a coding formula $\varphi_S(x, \bar{\mathbf{p}})$ is Σ_3^A as long as the parameters $\bar{\mathbf{q}}$ for the nice effective successor structure determining the domain of the model and $\mathbf{g}_0, \mathbf{g}_1$ are recursive in A.

We now claim that $\mathbf{x} \in \mathbf{L}_2$ if and only for any such $\bar{\mathbf{q}}, \mathbf{g}_0, \mathbf{g}_1 \leq_T \mathbf{x}$ the set S coded by $\mathbf{g}_0, \mathbf{g}_1$ is Σ_3. Moreover, this property is definable in $\mathcal{D}(\leq\mathbf{0}')$ and so proves the Theorem.

First suppose that $\mathbf{x} \in \mathbf{L}_2$. Then our initial remarks show that $S \in \Sigma_3^X$ for any $X \in \mathbf{x}$. As $X'' \equiv_T 0''$, $\Sigma_3^X = \Sigma_3$ by Proposition 8.4.2. Next, if $\mathbf{x} \notin \mathbf{L}_2$, then by Exercise **??** and Theorem 8.3.3 there are parameters $\bar{\mathbf{q}}$ defining a nice effective successor model with join $\mathbf{c} < \mathbf{x}$ with $\mathbf{c}' = \mathbf{0}'$. By Theorem 7.3.4, we can extend these parameters to ones $\bar{\mathbf{p}}$ defining a standard model of arithmetic which, of course, satisfies the definable properties guaranteeing that it is such a model. Now, by Theorem 8.3.17, for any $S \in \Sigma_3^X$ there are $\mathbf{g}_0, \mathbf{g}_1 \leq_T \mathbf{x}$ which code S in this model. Since $\mathbf{x}'' > \mathbf{0}''$ there is an $S \in \Sigma_3^X - \Sigma_3$ again by Proposition 8.4.2 and so a code for such an S below $\mathbf{x}$ as required.

Finally, note that, as we are working in definable standard models of arithmetic, we can definably say that a set is Σ_3 simply by using the translation into our degree structure of the corresponding sentence of arithmetic. ■

efdjclasses **Theorem 8.4.4** *For every $\mathbf{h} \geq \mathbf{0}''$ which is r.e. in $\mathbf{0}''$, the set $\{\mathbf{x} \leq \mathbf{0}' | \mathbf{x}'' = \mathbf{h}\}$ is definable in $\mathcal{D}(\leq\mathbf{0}')$.*

Proof. The previous theorem handles the case that $\mathbf{h} = \mathbf{0}''$. For $\mathbf{h} > \mathbf{0}''$ Let $E \in \mathbf{e} \in [\mathbf{0}', \mathbf{0}'']$ be such that $E' \in \mathbf{h}$. There is such an E by Corollary 8.3.11 and we can fix a definition of one in arithmetic. Consider the formula which says that for any $\mathbf{q}, \mathbf{g}_0, \mathbf{g}_1 < \mathbf{x}$ and $\bar{\mathbf{p}}$ which define a standard model of arithmetic and a set S coded in the model as in the proof of the Theorem, $S \in \Sigma_2^E$ and for any set $\hat{S} \in \Sigma_2^E$ (again as given by a definition in arithmetic) there are such $\mathbf{q}, \mathbf{g}_0, \mathbf{g}_1 < \mathbf{x}$ and $\bar{\mathbf{p}}$ defining $\hat{S}$. Proposition 8.4.2 and calculations already described now show that this guarantees that $\Sigma_2^{X'} = \Sigma_3^X = \Sigma_2^E$ and so $\mathbf{x}'' = \mathbf{e}' = \mathbf{h}$ as required. ■

defjclasses **Corollary 8.4.5** *The jump classes $\mathbf{L}_n$ ($\mathbf{a}^{(n)} = \mathbf{0}^{(n)}$) and $\mathbf{H}_n$ ($\mathbf{a}^{(n)} = \mathbf{0}^{(n+1)}$) are definable in $\mathcal{D}(\leq\mathbf{0}')$ for $n \geq 2$.*

Proof. In the proof of Theorem 8.4.4, require instead of $E' \in \mathbf{h}$ that $E^{(n-1)} \equiv_T 0^{(n)}$ for $\mathbf{L}_n$ and $E^{(n-1)} \equiv_T 0^{(n+1)}$ for $\mathbf{H}_n$. ■

By a separate additional argument that requires results beyond the scope of these lectures we can also get the definability of $\mathbf{H}_1$. While we could make such an argument at this point it will be easier later. We do so in Corollary defH1 8.4.11. The definability of $\mathbf{L}_1$ in $\mathcal{D}(\leq\mathbf{0}')$ is an important open problem.

If we now wish to deal with arbitrary relations on $\mathcal{D}(\leq\mathbf{0}')$ rather than simply subsets, we are faced with the problem that our analysis so far has, for each degree $\mathbf{a}$, produced various models of arithmetic in which we code the sets Σ_3^A. To discuss even binary relations we must have a way to analyze any $\mathbf{a}$ and $\mathbf{b}$ (or equivalently the sets coded below them as long as we are only working up to invariance under the double jump) in a single model (perhaps with additional correctness conditions). The basic formulation of this issue is given by asking about the biinterpretability of the structure (here $\mathcal{D}(\leq\mathbf{0}')$) with arithmetic (here first order). A similar notion applies to other structures (such as the r.e. degrees, $\mathcal{R}$) still with first order arithmetic and to ones such as $\mathcal{D}$ but for second order arithmetic.

biint **Definition 8.4.6** *A degree structure $\mathcal{S}$ is* biinterpretable with true first (second) order arithmetic *if it is interpretable in first (second) order arithmetic and we have formulas in parameters $\bar{\mathbf{p}}$ (including a correctness condition) as specified in* interp *§7.1 which provide an interpretation of true arithmetic in $\mathcal{S}$ (i.e. the models $\mathcal{M}(\bar{\mathbf{p}})$ satisfying the correctness condition are all standard). For second order arithmetic, we also have a formula $\varphi_S(x, \bar{y})$ which defines sets (coded) in the model given by $\bar{\mathbf{p}}$. We require that the sets defined by $\varphi_S(x, \bar{y})$ as $\bar{y}$ ranges over all parameters in $\mathcal{S}$ are all subsets of $\mathbb{N}$.*

Moreover, for both first and second order arithmetic, there is an additional formula $\varphi_R(x, \bar{y}, \bar{\mathbf{p}})$ such that $\mathcal{S} \vDash \forall x \exists \bar{y} \varphi_R(x, \bar{y}, \bar{\mathbf{p}})$ and for every $\mathbf{a}, \bar{\mathbf{g}} \in \mathcal{S}$, $\mathcal{S} \vDash \varphi_R(\mathbf{a}, \bar{\mathbf{g}}, \bar{\mathbf{p}})$ if and only if the set $\{n | \varphi_S(\mathbf{d}_n, \bar{\mathbf{g}}, \bar{\mathbf{p}})\}$ (where $\mathbf{d}_n$ is the nth element of the model $\mathcal{M}(\bar{\mathbf{p}})$ coded by the parameters $\bar{\mathbf{p}}$) is of degree $\mathbf{a}$. These last conditions then say that the set coded in $\mathcal{M}(\bar{\mathbf{p}})$ by $\bar{\mathbf{g}}$ is of degree $\mathbf{a}$ and that all degrees $\mathbf{a}$ in $\mathcal{S}$ have codes $\bar{\mathbf{g}}$ for a set of degree $\mathbf{a}$.

We say that $\mathcal{S}$ is biinterpretable with true first (second) order arithmetic up to double jump *if we weaken the second condition on φ_R so that for every $\mathbf{a}, \bar{\mathbf{g}} \in S$, $\mathcal{S} \vDash \varphi_R(\mathbf{a}, \bar{\mathbf{b}}, \bar{\mathbf{p}})$ if and only if the set $\{n | \varphi_S(\mathbf{d}_n, \bar{\mathbf{g}}, \bar{\mathbf{p}})\}$ has the same double jump as $\mathbf{a}$.*

It is not hard to see that, if a degree structure $\mathcal{S}$ is biinterpretable with first or second order arithmetic, then we know all there is to know about definability in, and automorphisms of, $\mathcal{S}$.

biintdef **Theorem 8.4.7** *If a degree structure $\mathcal{S}$ is biinterpretable with first or second order arithmetic then it is rigid, i.e. it has no automorphisms other than the identity, and a relation on $\mathcal{S}$ is definable in $\mathcal{S}$ if and only if it is definable in first or second order arithmetic, respectively.*

Proof. We first prove rigidity. Let $\bar{\mathbf{p}}$ satisfy all the formulas required for it to determine a standard model of arithmetic via the given formulas. Consider any $\mathbf{a} \in \mathcal{S}$ with some $\bar{\mathbf{g}}$ such that $\mathcal{S} \vDash \varphi_R(\mathbf{a}, \bar{\mathbf{g}}, \bar{\mathbf{p}})$ and any automorphism Ψ of $\mathcal{S}$. The image $\Psi(\bar{\mathbf{p}}) = \bar{\mathbf{r}}$ satisfies all the same formulas as $\bar{\mathbf{p}}$ and so also defines a standard model of arithmetic. The image $\bar{\mathbf{h}}$ of $\bar{\mathbf{g}}$ under Ψ also determines a subset of this model via φ_S and it must be the "same" subset in the sense that they correspond to the same subset of $\mathbb{N}$ via the isomorphisms among $\mathcal{M}(\bar{\mathbf{p}})$, $\mathcal{M}(\bar{\mathbf{r}})$ and $\mathbb{N}$. Of course, $\varphi_R(\mathbf{b}, \bar{\mathbf{h}}, \bar{\mathbf{r}})$ (where $\mathbf{b} = \Psi(\mathbf{a})$) is also true in $\mathcal{S}$ since Ψ is an automorphism. Our definition of biinterpretability now says that $\mathbf{a} = \mathbf{b}$ as required for rigidity.

Now consider any relation $Q(\bar{x})$ on $\mathcal{S}$. By the assumption that $\mathcal{S}$ is interpretable in first or second order arithmetic, we know that Q is definable in those structures. For the other direction, suppose Q is definable by a formula Θ of first or second order arithmetic. If this is first order arithmetic then we expanded it by a sequence $\bar{X}$ of second order parameters (of the same length n as $\bar{x}$) whose intended interpretations are some subsets of the model. If it is second order arithmetic then we simply assume that the formula already contains a sequence $\bar{X}$ of free second order variables (of the same length as $\bar{x}$). In any case, Θ defines the property that the sequence of the degrees of X satisfies Q.) Q is then defined in $\mathcal{S}$ by the formula $\Psi(\bar{z}) \equiv \exists \bar{p}, \bar{g}_0 \ldots \exists \bar{g}_{n-1}(\varphi_c(\bar{p}) \ \& \ \bigwedge_{i<n} \varphi_R(z_i, \bar{g}_i, \bar{p}) \rightarrow \Theta^T(\bar{g}_i, \bar{p}))$ where T is the translation of formulas of second order arithmetic given in §interp 7.1. Here our correctness condition φ_c guarantees that the model $\mathcal{M}(\bar{p})$ is standard and we also assume that the requirements of the definition of biinterpretability are satisfied. So the translation of Θ asserts (because of the properties of φ_R) that a sequence of sets of degree z_i satisfy Θ (in $\mathbb{N}$), i.e. Q holds of $\bar{z}$. ■

Our goal now is to prove that $\mathcal{D}(\leq \mathbf{0}')$ is biinterpretable with arithmetic up to double jump and so every relation on it invariant under the double jump is definable in it if and only if it is definable in first order arithmetic.

DObiintDJ **Theorem 8.4.8** *$\mathcal{D}(\leq \mathbf{0}')$ is biinterpretable with arithmetic up to double jump.*

Theorem defL2 8.4.3 and Theorem defdjclasses 8.4.4 show that we can define the double jump classes of degrees $\mathbf{a}$ in $\mathcal{D}(\leq \mathbf{0}')$ by talking about the sets that are coded (by our usual formula $\varphi_S(x, \bar{\mathbf{g}})$) in standard models $\mathcal{M}(\bar{\mathbf{p}})$ of arithmetic with $\bar{\mathbf{q}}, \bar{\mathbf{g}}$ below $\mathbf{a}$ as in the proof of Theorem defL2 8.4.3. The point here is that these sets determine Σ_3^A and so $\mathbf{a}''$ by Proposition sigma3=dj 8.4.2. If we wish to define the relations needed for biinterpretability up to double jump, we need to be able to talk about the sets that are Σ_3^A for an arbitrary degree $\mathbf{a}$ simultaneously in a single model. Our plan is to provide a scheme defining isomorphisms between two arbitrary standard models satisfying some additional correctness condition. Such isomorphisms would allow us to definably transfer (codes for) sets in different models to ones for the same sets in a single model and so define the required relation φ_R. We begin with a lemma that is used to build such isomorphisms by interpolating a sequence of additional models between the two given ones and isomorphisms between each successive pair of models.

enovertwoL1 **Lemma 8.4.9** *If $\mathbf{c} \leq \mathbf{0}'$, $\mathbf{c} \in \overline{\mathbf{L}}_2$, $\mathbf{a}_0, \mathbf{a}_1 \in \mathbf{L}_1$ and $\mathcal{P}$ is a recursive notion of forcing, then there is a $G \leq_T C$ which is 1-generic for $\mathcal{P}$ and such that $A_0 \oplus G$ and $A_1 \oplus G$ are both low.*

Proof. Let $D_{n,2}$ be the usual dense sets for making G 1-generic for $\mathcal{P}$. They, and the density function for them, are uniformly recursive in $0'$. Now consider, for $i \in \{0,1\}$, the sets $D_{n,i} = \{p | \Phi_n^{A_i \oplus V(p)}(n) \downarrow$ or $(\forall q \leq p)\Phi_n^{A_i \oplus V(q)}(n) \uparrow\}$. As the A_i are low, these sets and their density functions are also uniformly recursive in $0'$. Thus, by Theorem 8.3.9, there is a 1-generic sequence $\langle p_k \rangle$ and an associated generic set G both recursive in C meeting all these dense sets. Any such G clearly has all the properties required in the theorem. (Follow, for example, the proof of Proposition 6.2.18 using these $D_{e,i}$ in place of $D_{1,e}$.) ■

Proof (of Theorem 8.4.8). In addition to the previous correctness conditions for our standard models $\mathcal{M}(\bar{\mathbf{p}})$ we require for the rest of this section that $\mathbf{p}_0$, the first of the parameters $\bar{\mathbf{p}}$, which bounds the parameters $\bar{\mathbf{q}}$ defining the nice effective successor structure providing the domain $\mathbf{d}_n$ of the model, is in $\bar{\mathbf{L}}_2$. (This condition is definable by Theorem 8.4.3.) Given two such models $\mathcal{M}(\bar{\mathbf{p}}_0)$ and $\mathcal{M}(\bar{\mathbf{p}}_4)$ we want to show that there are additional models $\mathcal{M}(\bar{\mathbf{p}}_k)$ for $k \in \{1,2,3\}$ and uniformly definable isomorphisms between the domains of these models taking $\mathbf{d}_{i,n}$ to $\mathbf{d}_{1+1,n}$ for $i < 4$. (Given parameters $\bar{\mathbf{p}}_k$ defining a model $\mathcal{M}(\bar{\mathbf{p}}_k)$ we write $\mathbf{d}_{k,n}$ for the degree representing the nth element of this model. Similarly, we write $\bar{\mathbf{p}}_{k,0}$ for the first element of $\bar{\mathbf{p}}_k$ and $\bar{\mathbf{q}}_k$ for the parameters in $\bar{\mathbf{p}}_k$ determining the effective successor structure which provides the domain of $\mathcal{M}(\bar{\mathbf{p}}_k)$.) Thus (as we explain below) we produce a single formula $\theta(x, y, \bar{z}, \bar{z}')$ which uniformly defines isomorphisms between any two of our standard models $\mathcal{M}(\bar{\mathbf{p}}_0)$ and $\mathcal{M}(\bar{\mathbf{p}}_4)$ (with $\bar{z}$ and $\bar{z}'$ replaced by $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_4$).

We begin by choosing $\bar{\mathbf{q}}_1 < \mathbf{0}'$ as given by a 1-generic over $\mathbf{p}_{0,0}$ sequence and function for the recursive notion of forcing (Exercise ??) that embeds a nice effective successor model with $\bar{\mathbf{q}}_{1,0}$, the first element of $\bar{\mathbf{q}}_1$, being the bound on all the other required parameters. As $\mathbf{p}_{0,0} \in \mathbf{L}_2$, $\mathbf{0}'$ is $\bar{\mathbf{L}}_2(\mathbf{p}_{0,0})$ and so such $\bar{\mathbf{q}}_1$ exists by Theorem 8.3.3 (relativized to $\mathbf{p}_{0,0}$). Note that $\bar{\mathbf{q}}_1$ (and so $\bar{\mathbf{q}}_{1,0}$) is in $\mathbf{L}_1$ by Proposition 6.2.18 as it is associated with a 1-generic sequence recursive in $\mathbf{0}'$. We may now extend $\bar{\mathbf{q}}_1$ to $\bar{\mathbf{p}}_1$ defining a standard model $\mathcal{M}(\bar{\mathbf{p}}_1)$ by Exercise 7.3.3 and Theorem 8.3.3 as $\mathbf{0}'$ is $\overline{\mathbf{GL}}_2(\bar{\mathbf{q}}_1)$. Similarly, we see that there are $\bar{\mathbf{q}}_3$ and $\bar{\mathbf{p}}_3$ bearing the same relation to $\mathcal{M}(\mathbf{p}_4)$ as $\bar{\mathbf{q}}_1$ and $\bar{\mathbf{p}}_1$ do to $\mathcal{M}(\mathbf{p}_0)$. Now as $\bar{\mathbf{q}}_{1,0}$ and $\bar{\mathbf{q}}_{3,0}$ are both low we may apply Lemma 8.4.9 to the forcing of Exercise ?? to get $\bar{\mathbf{q}}_2 < \mathbf{0}'$ (again as $\mathbf{0}' \in \bar{\mathbf{L}}_2(\bar{\mathbf{q}}_{1,0}), \bar{\mathbf{L}}_2(\bar{\mathbf{q}}_{3,0})$) such that both $\bar{\mathbf{q}}_{1,0} \oplus \bar{\mathbf{q}}_{2,0}$ and $\bar{\mathbf{q}}_{2,0} \oplus \bar{\mathbf{q}}_{3,0}$ are in $\mathbf{L}_1$ and then extend $\bar{\mathbf{q}}_2$ to $\bar{\mathbf{p}}_2$ defining $\mathcal{M}(\bar{\mathbf{p}}_2)$ as we did for $\bar{\mathbf{q}}_1$.

We now apply Exercise 7.3.3 and Theorem 8.3.3 to get the desired schemes defining our desired isomorphisms: Given any $n \in \mathbb{N}$ and $i < 4$, consider the finite sequences of degrees $\langle \mathbf{d}_{i,0}, \ldots, \mathbf{d}_{i,n} \rangle$ and $\langle \mathbf{d}_{i+1,0}, \ldots, \mathbf{d}_{i+1,n} \rangle$. We want to show that there are parameters $\bar{\mathbf{r}}_i < \mathbf{0}'$ such that the formula $\varphi_2(x, y, \bar{\mathbf{r}}_i)$ (where $\varphi_2(x, y, \bar{z})$ ranges over binary relations as $\bar{z}$ varies as in Theorem 7.2.3) defines an isomorphism taking $\mathbf{d}_{i,k}$ to $\mathbf{d}_{i+1,k}$ for each $k \leq n$. By the results just cited it suffices to show that the $\bigoplus_{k<n} \mathbf{d}_{i,k} \oplus \bigoplus_{k<n} \mathbf{d}_{i+1,k}$ are in $\mathbf{L}_2$ for each $i < 4$. For $i = 0$, note that $\bar{\mathbf{q}}_1$ is associated with a 1-generic/$\mathbf{p}_{0.0}$ sequence which is recursive in $\mathbf{0}'$. Thus by Proposition 6.2.18 (suitably relativized) $(\bar{\mathbf{q}}_1 \oplus \mathbf{p}_{0,0})' = \mathbf{p}_{0,0}'$ and so $(\bar{\mathbf{q}}_{1,0} \oplus \bar{\mathbf{p}}_{0,0})' = \mathbf{p}_{0,0}'$. As $\mathbf{p}_{0,0} \in \mathbf{L}_2$, $\mathbf{0}'' = \bar{\mathbf{p}}_{0,0}'' = (\bar{\mathbf{q}}_{1,0} \oplus \bar{\mathbf{p}}_{0,0})''$ as required. The argument for $i = 3$ is similar. For the other pairs, we have already guaranteed that $\bar{\mathbf{q}}_{1,0} \oplus \bar{\mathbf{q}}_{2,0}$ and $\bar{\mathbf{q}}_{2,0} \oplus \bar{\mathbf{q}}_{3,0}$ are both $\mathbf{L}_1$.

We can now define the desired isomorphism $\theta(\mathbf{n}, \mathbf{m}, \bar{\mathbf{p}}_0, \bar{\mathbf{p}}_4)$ between $\mathcal{M}(\bar{\mathbf{p}}_0)$ and $\mathcal{M}(\bar{\mathbf{p}}_4)$. We say that an $\mathbf{n}$ in the domain of $\mathcal{M}(\bar{\mathbf{p}}_0)$ (i.e. $\varphi_D(\mathbf{n}, \bar{\mathbf{p}}_0)$) is taken to $\mathbf{m}$ in the domain of $\mathcal{M}(\bar{\mathbf{p}}_4)$ if and only if there are degrees $\bar{\mathbf{p}}_k$ for $k \in \{1, 2, 3\}$ defining models of arithmetic $\mathcal{M}(\bar{\mathbf{p}}_k)$ and ones $\bar{\mathbf{r}}_i$ for $i < 4$ as above such that each $\varphi_2(x, y, \bar{\mathbf{r}}_i)$ defines an isomorphism between initial segments of (the domains of) $\mathcal{M}(\bar{\mathbf{p}}_i)$ and $\mathcal{M}(\bar{\mathbf{p}}_{i+1})$ where the initial segment in $\mathcal{M}(\bar{\mathbf{p}}_0)$ is the one with largest element $\mathbf{n}$ and that in $\mathcal{M}(\bar{\mathbf{p}}_1)$ has largest element $\mathbf{m}$. Clearly this can all be expressed using the formulas $\varphi_D(x, \bar{\mathbf{p}}_k)$ and $\varphi_<(x, y, \bar{\mathbf{p}}_k)$ defining the domains of $\mathcal{M}(\bar{\mathbf{p}}_k)$ and the orderings on them. Note that the definition of this isomorphism is uniform in $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_4$ and that we have shown that for any $\bar{\mathbf{p}}_0$ and $\bar{\mathbf{p}}_4$ defining our standard models of arithmetic, there are parameters below $\mathbf{0}'$ defining all these isomorphisms. In other words, we have described the desired formula $\theta(x, \bar{y}, \bar{z}, \bar{z}')$.

We now wish to define the formula $\varphi_R(x, \bar{y}, \bar{\mathbf{p}}_0)$ required in the definition of biinterpretability up to double jump (for $\mathcal{M}(\bar{\mathbf{p}}_0)$ a model of arithmetic). (We have replaced $\bar{\mathbf{p}}$ in Definition 8.4.6 by $\bar{\mathbf{p}}_0$ to match our current notation.) First, φ_R says that, if $x \in \mathbf{L}_2$ (as defined by Theorem 8.4.3), then $\bar{y}$ defines (via our standard φ_S) the empty set in $\mathcal{M}(\bar{\mathbf{p}}_0)$. In addition, φ_R says that, if $x \notin \mathbf{L}_2$ and S is the set defined in $\mathcal{M}(\bar{\mathbf{p}}_0)$ by $\bar{y}$, then for every set $\hat{S} \in \Sigma_3^S$ (with $\hat{S}$ defined by other parameters $\bar{\mathbf{h}}$ in $\mathcal{M}(\bar{\mathbf{p}}_0)$ and $\hat{S} \in \Sigma_3^S$ expressed in the translation of arithmetic into $\mathcal{M}(\bar{\mathbf{p}}_0)$), there are $\bar{\mathbf{g}} < \mathbf{x}$ and $\bar{\mathbf{p}}_4$ with $\bar{\mathbf{p}}_{4,0} < \mathbf{x}$ such that $\bar{\mathbf{g}}$ codes a set $\hat{S}_4$ in $\mathcal{M}(\bar{\mathbf{p}}_4)$ and, for every $\mathbf{n}$ and $\mathbf{m}$, $\theta(\mathbf{n}, \mathbf{m}, \bar{\mathbf{p}}_0, \bar{\mathbf{p}}_4)$ implies that $\varphi_S(\mathbf{n}, \bar{\mathbf{h}}, \bar{\mathbf{p}}_0) \Leftrightarrow \varphi_S(\mathbf{m}, \bar{\mathbf{g}}, \bar{\mathbf{p}}_4)$, i.e. $\hat{S} = \hat{S}_4$. By all that we have done already, this guarantees that every $\hat{S} \in \Sigma_3^S$ is Σ_3^X. For the other direction, φ_R also says that if $\bar{\mathbf{g}} < \mathbf{x}$ and $\bar{\mathbf{p}}_4$ with $\bar{\mathbf{p}}_{4,0} < \mathbf{x}$ are such that $\bar{\mathbf{g}}$ codes a set $\hat{S}_4$ in $\mathcal{M}(\bar{\mathbf{p}}_4)$ then there is a set $\hat{S}$ (coded in $\mathcal{M}(\bar{\mathbf{p}}_0)$ by some $\bar{\mathbf{h}}$) which is Σ_3^S (as expressed in $\mathcal{M}(\bar{\mathbf{p}}_0)$) such that $\hat{S} = \hat{S}_4$ as expressed as above using θ. So again by what we have already done, this guarantees that every $\hat{S}_4 \in \Sigma_3^X$ is Σ_3^S. Thus, by Proposition 8.4.2, S has the same double jump as X as required. ■

D0defDJ **Theorem 8.4.10** *A relation on $\mathcal{D}(\leq \mathbf{0}')$ which is invariant under the double jump is definable in $\mathcal{D}(\leq \mathbf{0}')$ if and only if it is definable in true first order arithmetic.*

Proof. Follow the proof of Theorem 8.4.7 but use Theorem 8.4.8 in place of the assumption that the structure is biinterpretable with arithmetic. ■

defH1 **Corollary 8.4.11** *$\mathbf{H}_1$ is definable in $\mathcal{D}(\leq \mathbf{0}')$.*

Proof. This follows immediately from the Theorem and fact that $\mathbf{x} < \mathbf{0}'$ is in $\mathbf{H}_1$ if and only if $\mathcal{D}(\leq \mathbf{0}') \vDash \forall z \exists y \leq x(z'' = y'')$. This fact is proven for r.e. $\mathbf{x}$ in Nies, Shore and Slaman [1998, Theorem 2.21] but (as indicated there on p. 257) replacing the last use of the Robinson jump interpolation theorem in the proof by Theorem 8.3.10 provides a proof for $\mathcal{D}(\leq \mathbf{0}')$. ■

The analogous theorems hold for both $\mathcal{D}$ and $\mathcal{R}$, i.e. they are biinterpretable with second or first order arithmetic, respectively, up to double jump. (Moreover, in $\mathcal{D}$ the jump is also definable.) Their definable relations which are invariant under the double jump are then characterized in the same way. Indeed, every jump ideal $\mathcal{I}$ of $\mathcal{D}$ (i.e. an ideal that is also closed under the jump operator) which contains $\mathbf{0}^{(\omega)}$ is biinterpretable with second order arithmetic up to double jump if one takes the second order structure to have sets precisely those with degrees in $\mathcal{I}$ and the jump is definable in $\mathcal{I}$ as well.

By more extensive uses of Theorem 8.3.9 we can prove our biinterpretability and so definability results for $\mathcal{D}(\leq \mathbf{x})$ for any $\mathbf{x} \leq \mathbf{0}'$ in $\bar{\mathbf{L}}_2$.

L2biintDJ **Exercise 8.4.12** *For every $\mathbf{x} \leq \mathbf{0}'$ in $\bar{\mathbf{L}}_2$, $Th(\mathcal{D}(\leq\mathbf{x})$ is biinterpretable with true first order arithmetic and so its theory is 1-1 equivalent to that of true arithmetic. Moreover, for every $\mathbf{x} \leq \mathbf{0}'$ a relation on $\mathcal{D}(\leq \mathbf{x})$ invariant under double jump is definable in $\mathcal{D}(\leq \mathbf{x})$ if and only if it is definable in first order arithmetic. (For $\mathbf{x} \in \mathbf{L}_2$, this last result is trivial. Otherwise, it follows from biinterpretability as before.)*

The *Biinterpretability Conjectures* for $\mathcal{D}(\leq\mathbf{0}')$, $\mathcal{R}$ and $\mathcal{D}$ assert that these structures are actually biinterpretable with first, first and second order arithmetic, respectively. As we have seen proofs of these conjectures would show that the structures are rigid and would completely characterize their definable relations. These are the major open problems of degree theory.

Notes: The definitions of biinterpretability for different degree structures and the associated conjectures are due to Harrington and Slaman and Woodin (see Slaman [1991] and [2008]). Theorems 8.4.3 and 8.4.4 are originally due to Shore [1988] but for triple jump in place of double jump. The improvement of one jump is essentially an application of Proposition 8.4.2 as pointed out in Nies, Shore and Slaman [1998] where Corollary 8.4.11 also appears. Slaman and Woodin also proved Theorem 8.4.7 (again see Slaman [1991] and [2008]). Plans for proving Theorem 8.4.8 were proposed in both Shore [1988] and more concretely in Nies, Shore and Slaman [1998] but neither actually provided the definitions of the required comparison maps nor the proofs that they exist as we have done here. Thus Theorems 8.4.8, 8.4.10 and the improvement to initial segments of $\mathcal{D}(\leq\mathbf{0}')$ bounded by any $\mathbf{x} \in \bar{\mathbf{L}}_2$ of Exercise 8.4.12 are new. A proof of Exercise 8.4.12 is in Shore [2014]. Biinterpretability up to double jump for the r.e. degree $\mathcal{R}$ is proven in Nies, Shore and Slaman [1998]. Slaman and Woodin (see Slaman [1991] and [2008]) proved it for $\mathcal{D}$. A very different proof that also gives the results described for jump ideals containing $\mathbf{0}^{(\omega)}$ is in Shore [2007]. The definability of the jump is proven in Shore and Slaman [1999] based on the results of Slaman and Woodin. This reliance is removed in Shore [2007] where the jump is also defined in every jump ideals containing $\mathbf{0}^{(\omega)}$. ??forward references??

8.5 Array Nonrecursive Degrees

The notion of array nonrecursiveness was originally introduced in the context of r.e. degrees to capture certain types of arguments in which one needed multiple permissions from (changes in) a given r.e. set to construct a desired set. (DJS I) It was phrased in terms of the r.e. set meeting (intersecting) the elements of certain types of arrays of uniformly given finite sets. It was later (DJS II) generalized to all degrees with a definition based on a domination property involving functions weak truth-table reducible to $0'$ and shown to have many of the properties of $\overline{\mathbf{GL}}_2$ degrees.

anrdeg **Definition 8.5.1** *A degree* $\mathbf{a}$ *is* **ANR** *if for every function* $g \leq_{wtt} 0'$ *there is an* $f \leq_T \mathbf{a}$ *such that* f *is not dominated by* g.

Exercise 8.5.2 *If* $\mathbf{a} \in \overline{\mathbf{GL_2}}$ *then* $\mathbf{a} \in$ **ANR**.

This notion is actually equivalent to two related ones, one seemingly weaker and the other seemingly stronger. (DJS and CSh)

anreq **Proposition 8.5.3** *The following are equivalent for a degree* $\mathbf{a}$*:*

1. $\mathbf{a}$ is **ANR**.
2. There is a function $f \leq_T \mathbf{a}$ which is not dominated by the least modulus function m_K for $0'$.
3. For any $A \in \mathbf{a}$ and $g = \Phi_e(A \oplus 0')$ such that there is a function $r \leq_T A$ bounding the use from $0'$ in the computation of g at each x, there is a $k \leq_T A$ which is not dominated by g.

Proof. That (1) implies (2) and (3) implies (1) are immediate from the definitions. We prove that (2) implies (3).

Without loss of generality we may assume that f, g and r are increasing. We define the required $k \leq_T A$ as follows: To calculate $k(n)$ compute, for each $s > n$ in turn, $\Phi_{e,fr(s)}(A \oplus 0'_{fr(s)}; n)$ (i.e. compute $fr(s)$ many steps in the standard enumeration of $0'$ and then, using this set as the second component of the oracle (and A for the first), compute Φ_e at n for $fr(s)$ many steps) until the computation converges and then add 1 to get the value of $k(n)$. This procedure must converge as $\Phi_e(A \oplus 0'; n)$ converges. Now, as m_K does not dominate f, there are infinitely many n such that there is a $j \in [r(n), r(n+1))$ with $f(j) > m_K(j)$. For such n we have $fr(n+1) > f(j) > m_K(j) \geq m_K r(n)$. Thus $0'_{fr(s)} \upharpoonright r(n) = 0' \upharpoonright r(n)$ for every $s > n$. So the computation of $\Phi_e(A \oplus 0'; n)$ is, step by step, the same as that of $\Phi_e(A \oplus 0'_{fr(s)}; n)$ for each $s > n$ as all the oracles agree on the actual use of the true computation. So eventually we get an $s > n$ such that $\Phi_{e,fr(s)}(f \oplus 0'_{fr(s)}; n) \downarrow$ and the output must be $\Phi_e(A \oplus 0'; n)$. Thus, for these n, $k(n) = g(n) + 1 > g(n)$ as required. ■

lowanr **Exercise 8.5.4** *There is an* $\mathbf{a} \in \mathbf{ANR}$ *with* $\mathbf{a} \in \mathbf{L}_1$*. In fact, there is a Cohen* 1*-generic* A *whose degree is* **ANR***. Hint: use Proposition 8.5.3(2) and the principal function ??definition?? of* A*.*

Exercises on f is ANR, relativizations and uniformity

That there are Cohen 1-generics below every $\mathbf{a} \in \mathbf{ANR}$ follows immediately from the proof of Theorem 8.3.3 and Remark 8.3.4. This, as usual, gives one whole array of corollaries. We now prove the analog for **ANR** of the stronger version given for $\overline{\mathbf{GL}}_2$ degrees in Theorem 8.3.9. This allows us to carry out almost all of the known forcing constructions for $\overline{\mathbf{GL}}_2$ degrees for **ANR** ones.

anrgenseq **Theorem 8.5.5** *If* A *is of* **ANR** *degree,* $\mathcal{P}$ *is an* A*-recursive notion of forcing,* $\mathcal{C} = \langle D_n \rangle$ *a sequence of sets dense in* $\mathcal{P}$ *(including the ones* $\{p| \, |V(p)| > l\}$ *for each* l*) with a density function* $d(x,y) = \Psi(A \oplus 0'; x, y)$ *such that the use from* $0'$ *in the computation of* $\Psi(A \oplus 0'; x, y)$ *is bounded by a function* $\hat{r} \leq_T A$*, then there is a* $\mathcal{C}$*-generic sequence* $\langle p_s \rangle$ *recursive in* A*. Indeed,* $\forall n \exists s (p_{s+1} = d(p_s, n))$*.*

Proof. Without loss of generality we may assume that $\hat{r}(x,y)$ is increasing in both x and y. Next note that the nondecreasing function $m_K \hat{r}(s,s)$ satisfies the hypotheses of Proposition 8.5.3(3), i.e. it is computable from $A \oplus 0'$ and its $0'$ use is bounded by a function $(\hat{r}(s,s))$ recursive in A. Finally note that the maximum of the running times of $\Psi(A \oplus 0'; x, y)$ for $x, y \leq s$ is also is such a function. (We run Ψ on each input and then output the sum of the number of steps needed to converge.) Finally, we let r be the maximum of these three functions so it too is of the desired form. By Proposition 8.5.3, we now have an increasing function $g \leq_T A$ not dominated by r. We use g to construct the desired generic sequence p_s by recursion.

We begin with $p_1 = \mathbf{1}$. At step $s+1$ we have (by induction) a nested sequence $\langle p_i | i \leq s \rangle$ with $p_i \leq s$. We calculate $0'_{g(s+1)}$ and see if there are any changes on the use from $0'$ in a computation based on which some D_m was previous declared satisfied. If so, we now declare it unsatisfied. Suppose n is the least $m < s+1$ such that D_m is not now declared satisfied. (There must be one as we declare at most one m to be satisfied at every stage and none at stage 1.) We compute $\Psi_{g(s+1)}(A \oplus 0'_{g(s+1)}; p_s, n)$. If the computation does not converge or gives an output q such that $q > s+1$ or $q \nleq_{\mathcal{P}} p_s$ we end the stage and set $p_{s+1} = p_s$. Otherwise, we end the stage, declare D_n to be satisfied on the basis of this computation of the output q and set $p_{s+1} = q$. Of course, $\langle p_s \rangle \leq_T A$.

We now verify that$\langle p_s \rangle$ is $\mathcal{C}$-generic and indeed $\forall n \exists s (p_{s+1} = d(p_s, n))$. Clearly if we ever declare D_n to be satisfied (and define p_{s+1} accordingly) and it never becomes unsatisfied again then $p_{s+1} = d(p_s, n)$. Moreover, if we ever declare D_n to be satisfied (and define p_{s+1} accordingly) and it remains satisfied at a point of the construction at which we have enumerated $0'$ correctly up to $r(p_s, n)$, then by definition $p_{s+1} = d(p_s, n)$ and D_n is never declared unsatisfied again. We now show that this happens.

Suppose all D_m for $m < n$ have been declared satisfied by s_0 and are never declared unsatisfied again. Let $s+1 \geq s_0$ be least such that $g(s+1) \geq r(s+1)$. If D_n was declared satisfied at some $t+1 \leq s$ on the basis of some computation of $\Psi_{g(t+1)}(A \oplus 0'_{g(t+1)}; p_t, n)$ and there is no change in $0'$ on the use of this computation by stage $g(s+1)$ then the computation is correct, $p_{t+1} = \Psi(A \oplus 0'; p_t, n) \in D_n$ and D_n is never declared unsatisfied again. (The point here is that by our choice of s, $g(s+1) > m_K r(s+1, s+1) \geq m_K r(p_t, n)$ and so $0'_{g(s)} \upharpoonright r(p_t, n) = 0' \upharpoonright r(p_t, n)$.) Otherwise, D_n is unsatisfied at s and the least such. By construction we compute $\Psi_{g(s+1)}(A \oplus 0'_{g(s+1)}; p_s, n)$. The definition of r along with our choice of g and s guarantee that this computation converges and is correct and so unless $d(p_s, n) > s+1$ we declare D_n satisfied, set $p_{s+1} = d(p_s, n)$ and D_n is never declared unsatisfied again. If $d(p_s, n) > s+1$, we set $p_{s+1} = p_s$ and, as D_n remains unsatisfied and the computations already found do not change, we continue to do this until we reach a stage $v+1 \geq d(p_s, n)$ at which point $p_v = p_s$ and we set $p_{v+1} = d(p_v, n)$ declare D_n satisfied and it is never unsatisfied again. ■

Exercise 8.5.6 *Prove that every recursive lattice $\mathcal{L}$ with 0 and 1 can be embedded in $\mathcal{D}(\leq\mathbf{a})$ preserving 0 and 1 for any $\mathbf{a} \in \mathbf{ANR}$. (DJS)*

Exercise 8.5.7 *Prove that every $\mathbf{a} \in \mathbf{ANR}$ has the cupping property.*

jump inversion others Exercises??

Our goal now is to characterize the **ANR** degrees as those degrees **a** such that every $\mathbf{b} \geq \mathbf{a}$ is **RRE**. We begin with the analog of Theorem gl2rre 8.3.15 which provides one half of the equivalence.

anrrre **Theorem 8.5.8** *If $\mathbf{a} \in \mathbf{ANR}$ then $\mathbf{a}$ is $\mathbf{RRE}$.*

Proof. We use an A-recursive notion of forcing $\mathcal{P}$ with conditions $p = \langle p_0, p_1, p_2 \rangle$, $p_i \in 2^{<\omega}$ such that

1. $|p_0| = |p_1|$, $p_0(d_n) = A(n-1)$, $p_1(d_n) = 1 - A(n-1)$ where d_n is n^{th} place where p_0, p_1 differ and
2. $(\forall e < |p_0 \oplus p_1|)(e \in p_0 \oplus p_1 \Leftrightarrow \exists x(\langle e, x \rangle \in p_2))$.

Extension in this notion of forcing is defined simply by $q \leq_{\mathcal{P}} p \Leftrightarrow q_i \supseteq p_i$ but note that this applies only to p and q in P. Membership in P and $\leq_{\mathcal{P}}$ are clearly recursive in A.

Our plan is to define a class $\mathcal{C}$ of dense sets D_n with a density function $d(p, n)$ recursive in $A \oplus 0'$ with $0'$ use recursively bounded. Theorem anrgenseq 8.5.5 then supplies a $\mathcal{C}$-generic sequence $\langle p_s \rangle \leq_T A$ from which we can define the required $G \leq_T A$ in which $\mathbf{a}$ is r.e. If $p_s = \langle p_{s,0}, p_{s,1}, p_{s,2} \rangle$ we let $G_i = \cup\{p_{s,i} | s \in \mathbb{N}\}$ for $i = 0, 1, 2$ so $G_i \leq_T A$. Then, if we can force G_0 and G_1 to differ at infinitely many places, $G_0 \oplus G_1 \equiv_T A$. On the other hand, the definition of the notion of forcing obviously makes $G_0 \oplus G_1$ r.e. in G_2. Thus $\mathbf{a}$ is be r.e. in $\mathbf{g} = \deg(G_2)$. We also have other requirements that make $\mathbf{g} < \mathbf{a}$ as well.

We begin with the dense sets that provide the differences we need:

$$D_{2n} = \{p \in \mathcal{P} : p_0, p_1 \text{ differ at at least } n \text{ points}\}.$$

We define the required function $d(r, 2n)$ by recursion on n. Given r and $n+1$, we suppose we have calculated $d(r, 2n) = p = \langle p_0, p_1, p_2\rangle \in D_{2n}$ with $p \leq_{\mathcal{P}} r$. If $p \notin D_{2n+2}$, we need to compute a $q = \langle q_0, q_1, q_2\rangle \in D_{2n+2}$ with $q \leq_{\mathcal{P}} p$. Let $q_0 = p_0\hat{\ }A(n)$, $q_1 = p_1\hat{\ }(1 - A(n))$. Choose $i \in \{0, 1\}$ such that $q_i(|p_0|) = 1$. Define $q_2 \supseteq p_2$ by choosing x large and setting $q_2(\langle 2|p_0| + i, x\rangle) = 1$ and $q_2(z) = 0$ for all $z \notin \mathrm{dom}(p_2)$ and less than $\langle 2|p_0| + i, x\rangle$. Now $q = \langle q_0, q_1, q_2\rangle$ satisfies the requirements to be a condition in P. It obviously extends p and is in D_{2n+2}. This computation is clearly recursive in A.

We must now add dense sets to guarantee that $A \not\leq_T G_2$:

$$\begin{aligned} D_{2n+1} &= \{p \in \mathcal{P} :\ \exists x(\Phi_n^{p_2}(x) \downarrow \neq A(x)) \text{ or } \forall(\sigma_0, \sigma_1 \supseteq p_2)[\exists x(\Phi_n^{\sigma_0}(x) \downarrow \neq \Phi_n^{\sigma_1}(x)) \downarrow \Rightarrow \\ (\exists i &\in \{0,1\})(\exists \langle e, x\rangle)(e < |p_0 \oplus p_1| \ \&\ \sigma_i(\langle e, x\rangle) = 1 \neq (p_0 \oplus p_1)(e)]\}. \end{aligned}$$

Of course, the first alternative guarantees that $\Phi_n^{G_2} \neq A$ while the second that $\Phi_n^{G_2}$, if total, is recursive. The point here is that if some p_s in our generic sequence satisfies the second clause then, we can, for any z, calculate $\Phi_n^{G_2}(z)$ by finding any $\sigma \supseteq p_{s,2}$ such that $\Phi_n^{\sigma}(z) \downarrow$ and taking its value as $\Phi_n^{G_2}(z)$. There is such a $\sigma \subseteq G_2$ as $\Phi_n^{G_2}$ is assumed to be total and $G_2 \supseteq p_{s,2}$. If there were some other $\tau \supseteq p_{s,2}$ with $\Phi_n^{\tau}(z) \downarrow \neq \Phi_n^{\sigma}(z) \downarrow$ then, by our choice of s and the definition of D_{2n+1}, there is no $\langle e, x\rangle$ with $e < |p_0 \oplus p_1|$ such that $\tau(\langle e, x\rangle) = 1 \neq (p_0 \oplus p_1)(e)$. Thus we could form a condition $q \leq_{\mathcal{P}} p_s$ with $q_2 = \tau$ by extending p_0 and p_1 by setting $q_1(w) = q_2(w) = 1$ (for $w \geq |p_0|$) if either $\langle 2w, v\rangle$ or $\langle 2w + 1, v\rangle$ is in τ for any v. In this way no new differences between q_0 and q_1 (not already in p_0 and p_1) occur and the definition of being a condition is satisfied. Thus q is a condition extending $p_{s,2}$ with $\Phi_n^{q_2}(z) \downarrow \neq A(z)$ contradicting our choice of s.

We compute the required density function $d(q, 2n+1)$ as follows. Given q we ask one question of $0'$ determined recursively in q: Are there extensions σ_0, σ_1 of q_2 that would show that q does not satisfy the second disjunct in the definition of D_{2n+1}. If not, let $d(q, 2n+1) = q$ which is already in D_{2n+1}. If so, we find the first such pair (appearing in a recursive search) and ask A which σ_i gives an answer different from $A(x)$. We now need a condition $r = d(q, 2n+1)$ extending q with third coordinate r_2 extending σ_i. For each $\langle e, x\rangle$ with $e \geq |q_1 \oplus q_2|)$ and $\sigma_i(\langle e, x\rangle) = 1$ we define $r_j(z) = 1$ for both $j \in \{0, 1\}$ for the z that makes $(r_0 \oplus r_1)(e) = 1$ and otherwise we let $r_j(u) = 0$ for all other u less than the largest element put into either r_0 or r_1 by the previous procedure. We now extend σ_i to the desired r_2 by putting in $\langle k, y\rangle$ for a large y for all those $k \geq |q_1|$ put into $r_0 \oplus r_1$ for which there is no $\langle k, w\rangle$ in σ_i. Otherwise we extend σ_i by 0 up to the largest element put in by this procedure. It is clear that this produces a condition r as required. (No points of difference between r_0 and r_1 are created that were not already present in q.)

We now apply Theorem anrgenseq 8.5.5 to get a $\mathcal{C}$-generic sequence $\langle p_s\rangle \leq_T A$. As promised, we let $G_i = \cup\{p_{s,i} | s \in \mathbb{N}\}$ for $i = 0, 1, 2$ and, as described above, $A \equiv_T G_0 \oplus G_1$ which is r.e. in G_2. In addition, the conditions in D_{2n+1} guarantee (as above) that $\Phi_n^{G_2} \neq A$ as well. ■

Exercise 8.5.9 *Prove that every* $\mathbf{a} \in \mathbf{ANR}$ *has the cupping property. Hint? Indifference set, i.e.* $f : \mathbb{N} \to \{0, 1, 2\}$ *approach??*

characterization as all above are RRE reference notions and terminology about trees from §spectormin 9.2

general definability issues forward refs

??Exercises on Relativization via Proposition mod ??.

anrreldef **Definition 8.5.10** *A function* f *is* ANR *if it is not dominated by* m. *It is* ANR *relative to* h *if* $h \leq_T f$ *and* f *is not dominated by* m_h. *A degree* $\mathbf{a}$ *is* **ANR** *relative to* $\mathbf{b}$, $\mathbf{ANR}(\mathbf{b})$, *if there are* $f \in \mathbf{a}$ *and* $h \in \mathbf{b}$ *such that* f *is* ANR *relative to* h, $ANR(h)$.

Chapter 9

mindegs

Minimal Degrees and Their Jumps

9.1 Introduction

We now return to extension of embeddings problem. We saw that as long as we do not attempt to put a new degree in the extension below a given degree, then anything consistent is possible (??Exercise extemb 5.2.22). We now turn toward the issue of whether one can put new degrees below given ones. The answer is strongly negative. In fact, strong enough so that we can rule out all the extensions not constructed by ??Exercise extemb 5.2.22 for finite lattices $\mathcal{P}$. Clearly embedding every finite lattice $\mathcal{P}$ as an initial segment of $\mathcal{D}$ suffices as then if $\mathcal{Q}$ adds elements below any of $\mathcal{P}$ then there can be no extension to $\mathcal{Q}$ of the embedding of $\mathcal{P}$ as an initial segment. We prove this and more in Chapter initialseg 10. This suffices to decide the truth of all two quantifier sentences in $\mathcal{D}$ (Chapter 2qtth 10.4) and also to show that the set of true three quantifier sentences is not decidable (Chapter 3qtth 10.5).

We begin with the simplest case.

Definition 9.1.1 *A degree* $\mathbf{a} > \mathbf{0}$ *is minimal if, for any* $\mathbf{b} \leq \mathbf{a}$*,* $\mathbf{b} = \mathbf{0}$ *or* $\mathbf{b} = \mathbf{a}$*. A degree is* $\mathbf{a}$ *is a minimal cover of* $\mathbf{c} > \mathbf{a}$ *if for any* $\mathbf{b}$ *with* $\mathbf{c} \leq \mathbf{b} \leq \mathbf{a}$*,* $\mathbf{b} = \mathbf{c}$ *or* $\mathbf{b} = \mathbf{a}$*.*

We cannot hope to construct a set of minimal degree by forcing with finite conditions like Cohen forcing as we have seen that generics for such forcings have every countable partial order embedded below them. We move then from approximations (conditions) that are clopen sets in Cantor space (all extensions of a $\sigma \in 2^{\mathbb{N}}$) to ones that are prefect subsets instead.

spectormin

9.2 Perfect forcing and Spector minimal degrees

We represent perfect subsets of Cantor space, $2^{\mathbb{N}}$ (i.e. nonempty sets with every point a limit point) by binary perfect (i.e. always branching) trees T (with no dead ends). The perfect subsets of Cantor space are then the paths $[T]$ through these trees. We present such trees as functions $T : 2^{<\omega} \to 2^{<\omega}$ with certain properties. ?? define Cantor space and relevant topology perfect trees etc. early on??

binfunctree **Definition 9.2.1** *A* binary function tree *is a (possibly partial) function* $T : 2^{<\omega} \to 2^{<\omega}$ *such that*

1. $\sigma \subseteq \tau \Rightarrow T(\sigma) \subseteq T(\tau)$ *(for* $\tau \in \mathrm{dom}(T)$*, so, in particular, if* $T(\tau)\downarrow$ *and* $\sigma \subseteq \tau$ *then* $T(\sigma)\downarrow$*) and*

2. $\sigma | \tau \Rightarrow T(\sigma) | T(\tau)$ *(for* $\sigma, \tau \in \mathrm{dom}(T)$*).*

Definition 9.2.2 *We say that a binary string* τ *is* on T *if there is a* σ *such that* $T(\sigma) = \tau$*. We say that* τ *is* on T above ρ *if there is a* $\sigma \supseteq \rho$ *with* $T(\sigma) = \tau$*.*

lth **Exercise 9.2.3** *If* T *is a binary function tree then (for* $\sigma \in \mathrm{dom}\, T$*),* $|T(\sigma)| \geq |\sigma|$*.*

Exercise 9.2.4 *If* T *is a binary tree in the sense of Definition* tree *4.2.1 then* $[T]$ *is perfect if and only if there is a total binary function tree* S *such that* $[S] = [T]$*. If* T *is recursive (as a subset of* $2^{<\omega}$*) then we may take* S *to be so as well.*

T[C] **Definition 9.2.5** *For total binary function trees, we let* $T[C] = \cup\{T(\sigma) | \sigma \subset C\}$ *for each set* C *and call it the* path in T determined by or following C*.*

Exercise 9.2.6 *If* T *is a total binary function tree then* $\{T[C] | C \in 2^{\mathbb{N}}\} = [T]$*, the set of paths in* T *as defined for general trees in Definition* tree *4.2.1.*

Exercise 9.2.7 *If* S *and* T *are total binary function trees then* $[S] \subseteq [T]$ *if and only if* $\forall\sigma\exists\tau(S(\sigma) \subseteq T(\tau))$*.*

Exercise 9.2.8 *If* S *is a total binary function tree and* $C \in 2^{\mathbb{N}}$*, then* $S[C] \equiv_T C \oplus S$*. If* T *is also a total binary function tree,* $[S] \subseteq [T]$ *and* $S[C] = T[D]$ *for some* $D \in 2^{\mathbb{N}}$*, then* $D \leq_T s[C] \oplus T$*.*

Exercise 9.2.9 *Prove there is a total binary tree* $T \leq_T 0'$ *such that* $\forall C(T[C]' \equiv_T C \vee 0')$ *and so if* $C \geq_T 0'$ *then* $T[C]' \equiv_T C$*. This provides a proof of the Friedberg Jump Inversion Theorem* frcomp *5.3.1 that exposes some of the uniformities in the proof.*

treeorder **Definition 9.2.10** *We define an order* $\leq_{\mathcal{S}}$ *on the binary function trees by* $S \leq_{\mathcal{S}} T \Leftrightarrow \forall\sigma(S(\sigma)\downarrow \Rightarrow \exists\tau(S(\sigma) = T(\tau))$*, in which case we say that* S *is a* subtree *of* T*.*

Remark 9.2.11 *In the remainder of this chapter all trees are partial recursive binary function trees (unless otherwise specified) and we just call them trees. In the rest of this section they are also total unless otherwise specified.*

Our forcing conditions, in this section, are these trees. The order relation $S \leq_{\mathcal{S}} T$ is then equivalent to $\forall\sigma\exists\tau(S(\sigma) = T(\tau))$.

The function V required in the definition of a notion of forcing is given by $V(T) = T(\emptyset)$ but the notion of extension makes it clear that the only possible generic sets G extending the condition T are the $G \in [T]$. This notion $\mathcal{S}$ of forcing with perfect recursive binary function trees is often called *Spector forcing.* Its analog in set theory is often called Sacks forcing or perfect forcing. Note that this notion of forcing is only recursive in $0''$. The crucial point here is that it takes $0''$ to determine if Φ_e is total. Once we know it is total, $0'$ suffices to determine if it is a binary function tree as this is then a Π_1 property. If S, T are conditions in $\mathcal{S}$ then $0'$ can also determine if $S \leq_{\mathcal{S}} T$ as this too is a Π_1 property. The point here is that if there is any τ such that $T(\tau) = S(\sigma)$ then it must be of length at most $|S(\sigma)|$ by Exercise lth 9.2.3

The requirements for a set G to be of minimal degree are as follows:

- N_e: $G \neq \Phi_e$ and
- M_e: If Φ_e^G is total then either Φ_e^G is recursive or $G \leq_T \Phi_e^G$.

The N_e requirements are very easy to meet.

spdiag **Lemma 9.2.12** *For each e the set of conditions $\{T|T \Vdash \neg(\Phi_e = G)\}$ is dense in $\mathcal{S}$. In fact the smaller set $D_e = \{T|\neg(\Phi_e = T(\emptyset)(x))\}$ is already dense in $\mathcal{S}$.*

Proof. Given any tree T and Φ_e, note that $\neg(T(i) = \Phi_e)$ for i at least one of 0 or 1 as $T(0)|T(1)$. Thus we may take as the desired extension S of T the subtree such that $S(\sigma) = T(i\hat{\ }\sigma)$, i.e. it starts with $T(i)$ for the appropriate i and then continues on as does T. ■

We formalize the operation that provides a witness to the density required in Lemma spdiag 9.2.12:

FuTree **Definition 9.2.13** *For any partial tree T and $\sigma \in 2^{<\omega}$, the* full subtree *of T above σ, $Fu(T, \sigma)$ or sometimes simply T_σ, is the tree S defined by $S(\tau) = T(\sigma\hat{\ }\tau)$.*

Proposition 9.2.14 *If T is (partial) recursive then so is T_σ and an index for it can be found uniformly recursively in one for T.*

Proof. Immediate. ■

Proposition 9.2.15 *There are density functions for the D_e of Lemma spdiag 9.2.12 which are uniformly recursive in $0'$ on the set of (recursive binary function) trees.*

Proof. Given any T, find an x such that $T(\sigma\hat{\ }0)(x) \neq T(\sigma\hat{\ }1)(x)$. Then ask $0'$ if $\Phi_e(x)\downarrow$. If so compute its value. In any case take $i \in \{0, 1\}$ such that $\neg(T(\sigma\hat{\ }i)(x) = \Phi_e(x))$ and take $Fu(T, \sigma\hat{\ }i)$ as the desired extension. ■

We must now see how to satisfy the minimality requirements M_e. We have seen several times how to make sure that Φ_e^G is recursive. To do this we want a be in a situation in which there are no extensions of the current approximation that e-split.

`nosplits` **Lemma 9.2.16** *If T is a partial tree such that there are no σ and τ such that $T(\sigma)|_e T(\tau)$, $G \in [T]$ and Φ_e^G is total then Φ_e^G is recursive.*

Proof. As usual, to compute $\Phi_e^G(x)$ we search for any σ such that $\Phi_e^{T(\sigma)}(x) \downarrow$. Since $\Phi_e^G(x) \downarrow$ there is an initial segment γ of G such that $\Phi_e^{\gamma}(x) \downarrow = \Phi_e^G(x)$. As $G \in [T]$ there is a τ such that $\gamma \subseteq T(\tau) \subset G$ and so τ is a string as desired. We then note that, for any such σ, $\Phi_e^{T(\sigma)}(x) = \Phi_e^{T(\tau)}(x) = \Phi_e^G(x)$ as otherwise $T(\sigma)|_e T(\tau)$. ■

We must now argue that if we cannot extend a given T to one with no e-splits on it as above, then we can guarantee that, if total, $\Phi_e^G \geq_T G$. To this end, we define another operation on trees that proceeds by searching for e-splits.

`esplittree` **Definition 9.2.17** *The* e-splitting subtree, *$Sp(T,e) = S$, of a partial recursive tree T is defined by recursion. $S(\emptyset) = T(\emptyset)$. If $S(\sigma) = T(\tau)$ then we search for $\tau_0, \tau_1 \supseteq \tau$ such that the $T(\tau_i)$ e-split. We let τ_0 and τ_1 be the first such pair found in a standard search and set $S(\sigma\hat{\ }i) = T(\tau_i)$. A partial recursive tree S is an* e-splitting tree *if, for every σ, if one of $S(\sigma\hat{\ }0)$, $S(\sigma\hat{\ }1)$ is convergent then both are and they form an e-split.*

`esplits` **Proposition 9.2.18** *$Sp(T,e)$ is a partial recursive subtree of T with an index given uniformly recursively in one for T. If $Sp(T,e)$ is not total then there is a τ such that there are no e-splits on T above τ for some τ. Indeed, if $Sp(T,e)(\hat{\tau}) \downarrow$ but $Sp(T,e)(\hat{\tau}\hat{\ }0) \uparrow$ and $T(\tau) = Sp(T,e)(\hat{\tau})$, then there are no e-splits on T above τ. Moreover, for any σ, $Sp(T,e)(\sigma\hat{\ }0) \downarrow \Leftrightarrow Sp(T,e)(\sigma\hat{\ }1) \downarrow$ and so $Sp(T,e)$ is an e-splitting tree.*

Proof. The assertions about the uniformity of the procedure of forming the e-splitting subtree and the equiconvergence of $Sp(T,E)(\sigma\hat{\ }i)$ for $i \in \{0,1\}$ are immediate from the definition. As for the rest, if $S(\emptyset) \uparrow$ then $T(\emptyset) \uparrow$ and we are done trivially. Otherwise, let τ be such that $Sp(T,e)(\tau) \downarrow = T(\rho)$ for some ρ but $Sp(T,e)(\tau\hat{\ }i) \uparrow$ for some (equivalently both) $i \in \{0,1\}$. If there were an e-splititng on T above ρ then we would have $S(\tau\hat{\ }i) \downarrow$ for both $i \in \{0,1\}$ by definition. ■

Thus to satisfy the minimality requirement M_e, it suffices to prove that if T_σ has e-splits for every σ (and so we cannot use Lemma $\overset{\texttt{nosplits}}{9.2.16}$ to force Φ_e^G to be recursive if total) then $Sp(T,e)$ forces $G \leq_T \Phi_e^G$ if the latter is total.

`complemma` **Lemma 9.2.19 (Computation Lemma)** *If S is a partial recursive e-splitting tree, $G \in [S]$ and Φ_e^G is total then $G \leq_T \Phi_e^G$.*

Proof. We compute an ascending sequence γ_n of initial segments of G (and so G itself) from Φ_e^G by recursion. We begin with $\gamma_0 = S(\emptyset)$ which is an initial segment of G since $G \in [S]$. Suppose we have $\gamma_n = S(\sigma_n) \subset G$. As $G \in [S]$, one of $S(\sigma_n\hat{\ }0)$ and $S(\sigma_n\hat{\ }1)$ is also an initial segment of G. Thus $S(\sigma_n\hat{\ }0)$ and $S(\sigma_n\hat{\ }1)$ are both convergent and e-split. We may then recursively find an x on which $\Phi_e^{S(\sigma_n\hat{\ }0)}(x) \downarrow \neq \Phi_e^{S(\sigma_n\hat{\ }1)}(x) \downarrow$. Exactly one of these two agrees with $\Phi_e^G(x)$. We choose that $i \in \{0,1\}$ and set $\sigma_{n+1} = S(\sigma_n\hat{\ }i)$. ■

We have thus proven the density of conditions needed to satisfy the minimality requirements.

itsoresplit **Lemma 9.2.20** *The sets $C_e = \{T|$ either there are no e-splits on T or T is an e-splititng tree$\}$ are dense in $\mathcal{S}$. Moreover, there are density functions for these sets which are uniformly recursive in $0''$ on the set of (recursive binary function) trees.*

Proof. By the above Lemmas, either there is a σ such that $Fu(T, \sigma)$ has no e-splits or $SP(T, e)$ is a total tree and so the C_e are dense. As the two options are Σ_2 and Π_2 properties, respectively, $0''$ can decide which option to take and, if the first is chosen then even $0'$ can find a suitable σ as there being no e-splits on T_σ is a Π_1 property. If the second is chosen then the index is given recursively. ■

spmindeg **Theorem 9.2.21** *There is a minimal degree $\mathbf{g} \leq \mathbf{0}''$. Indeed, for every degree $\mathbf{c}$ there is a $\mathbf{g} \leq \mathbf{c}''$ which is a minimal cover of $\mathbf{c}$.*

Proof. Take any generic sequence $\langle T_n \rangle$ meeting all the D_e and C_e. The associated generic set $G = \cup T_n(\emptyset)$ is of minimal degree. By the above results on the complexity of the density functions we may take the sequence and so G to be recursive in $0''$. The remark about minimal covers, now follows by simply relativizing the proof to $\mathbf{c}$. ■

Theorem spmindeg 9.2.21 says that every degree $\mathbf{c}$ has a minimal cover, and indeed one recursive in $\mathbf{c}''$. Our usual question at such a point is can we do better in in terms of the complexity of the minimal cover we construct (or equivalently in terms of the minimal degree the basic construction provides). Before turning to this line of analysis, we point to a deeper question: Which degrees are minimal covers? We provide a lot of information and partial answers to this question in Chapter PsJ ??. The analysis there will, in part, however, depend on the answer to the complexity question that we give in the next section.

The most obvious question about the complexity of minimal degrees is whether we can produce one below $\mathbf{0}'$. It seems clear that we cannot use Spector forcing for this as the notion of forcing (indeed even the set of conditions) is of degree $\mathbf{0}''$. Given the work that we have already done, however, one would try to use partial recursive trees instead. The basic lemmas that we have already proven (nosplits 9.2.16 and complemma 9.2.19) still work. The problem is that once we hit a partial tree, there may be no further extensions. We construct a sequence of trees that satisfy all the requirements and construct a minimal degree below $\mathbf{0}'$ in the next section. The crucial new facet of the construction is that we use partial trees but when we discover we have reached a terminal point we backtrack and revise the previous trees in our sequence. A priority argument is then needed to show that the sequence stabilizes and so we satisfy each requirement.

Another improvement that we can deal with in the setting of Spector forcing is saying something about the double jump of G. In particular, we can show that $G'' \equiv_T 0''$. As we have often seen, we can either introduce new dense sets (requirements) that directly control the double jump or cleverly argue that we have already done so. We present a direct proof an leave the indirect one as an exercise. The idea here is that $G'' \equiv_T Tot^G = \{e|\Phi_e^G$ is total$\}$ and so we want conditions that decide if $e \in Tot^G$. The route is similar to that taken to splitting trees. The first alternative is that we have a tree T and an x such that $\Phi_e^{T(\sigma)}(x) \uparrow$ for every σ. Obviously in this situation we have forced that $\Phi_e^G(x) \uparrow$

and so it is not total. The second alternative is to produce a tree T such that $\Phi_e^G(x)\downarrow$ for every x and every $G \in [T]$. The analog of the $Sp(T, e)$ is $Tot(T, e)$:

totdef **Definition 9.2.22** *If T is a (partial) tree then $S = Tot(T, e)$ is defined by recursion beginning with $S(\emptyset) = T(\emptyset)$. If we have $S(\sigma) = T(\tau)$ then search for a $\rho \supseteq \tau$ such that $\Phi_e^{T(\rho)}(|\sigma|)\downarrow$. If there is one we let ρ be the first found in a standard search and set $S(\sigma\hat{\ }i) = T(\rho\hat{\ }i)$ for $i \in \{0, 1\}$.*

Proposition 9.2.23 *An index for $Tot(T, e)$ can be found uniformly recursively in one for T. If $Tot(T, e)$ is not total then there is a σ and an x such that $\Phi_e^{T(\rho)}(x)\uparrow$ for every $\rho \supseteq \sigma$.*

Proof. This is immediate from the definition of $Tot(T, e)$. ■

totdense **Proposition 9.2.24** *The sets $B_e = \{T|\exists x\forall\sigma(\Phi_e^{T(\sigma)}(x)\uparrow)\}\cup\{T|(\forall\sigma)(\forall x < |\sigma|)(\Phi_e^{T(\sigma)}(x)\downarrow)\}$ are dense in $\mathcal{S}$ and uniformly recursive in $0''$ and so have density functions uniformly recursive in $0''$.*

Proof. To see that the B_e are dense consider any T. If $Tot(T, e)$ is a total function we have the desired extension. If not, then there is a σ and an x such that $\Phi_e^{T(\rho)}(x)\uparrow$ for every $\rho \supseteq \sigma$. So T_σ is then the desired extension. That the B_e are uniformly recursive in $0''$ is immediate from their definition. ■

Proposition 9.2.25 *If G is a generic defined from a sequence $\langle T_n\rangle \leq_T 0''$ meeting all the B_e then $G'' \equiv_T 0''$.*

Proof. To decide if $e \in Tot^G \equiv_T G''$, find an s such that $T_s \in B_e$ and see which clause of the definition of B_e is satisfied by T. ■

mintot **Theorem 9.2.26** *There is a minimal degree $\mathbf{g}$ with $\mathbf{g}'' = \mathbf{0}''$.*

Proof. Add the dense sets B_e to those C_e and D_e considered before. There is a generic sequence recursive in $0''$ meeting all these sets and the generic G associated with it has all the desried properties. ■

Of course, as might be naively expected, functions of minimal degree cannot have any strong domination properties. For example, none can be $\overline{\mathbf{GL}}_2$ by Theorem ngl2genseq ??. Even more striking is the fact that there is a single function of degree $\mathbf{0}'$ that dominates every function of minimal degree. This follows from the proof of Theorem ngl2genseq ??. In particular, by Proposition anreq 8.5.3 and Theorem anrgenseq 8.5.5 m_K, the least modulus function for $0'$ is such a function. For the minimal degrees we have constructed so far, we can say even more.

Exercise 9.2.27 *Show that the minimal degree constructed in Theorem mintot 9.2.26 is $\mathbf{0}$-dominated, i.e. every function recursive in G is dominated by a recursive function.*

Exercise 9.2.28 *Show that the G constructed in Theorem* mintot *9.2.26 has minimal tt and wtt degree. (Hint: Recall Exercise* 0domtt *8.1.2.)*

Exercise 9.2.29 *Show that the minimal degree constructed in Theorem* spmindeg *9.2.21 has double jump $\mathbf{0}''$. Hint: show that meeting the dense sets C_e guarantees that the sequence meets the B_e as well.*

posnerl

Exercise 9.2.30 (Posner's Lemma) *Show that meeting the dense sets C_e also guarantees that a generic sequence meets the D_e as well. Hint: Consider an n such that, for every σ and z, $\Phi_n^\sigma(z) = 0$ if $\neg(\exists x < |\sigma|)(\sigma(x) \neq \Phi_{e,|\sigma|}(x)\downarrow)$ and $\Phi_n^\sigma(z) = \sigma(z)$ otherwise.*

Exercise 9.2.31 *Show that for every $\mathbf{d} > \mathbf{0}$ there is a minimal degree $\mathbf{g} \leq_T \mathbf{d}' \vee \mathbf{0}''$ such that $\mathbf{g} \not\leq_{\mathbf{T}} \mathbf{d}$. ??Improvement in Exercise ??*

treeofmin

Exercise 9.2.32 *There are continuum many minimal degrees. Indeed, there is a binary function tree $T \leq_T 0''$ such that every $G \in [T]$ is of minimal degree. Hint: Use conditions (T, n) where T is a (recursive binary function) tree, $n \in \mathbb{N}$ and extension is defined by $(S, m) \leq (T, n)$ if $S \leq_S T$, $m \geq n$ and $S(\sigma) = T(\sigma)$ for every σ of length $\leq n$.*

ntantichain

Exercise 9.2.33 *Use the previous Exercise to show that any maximal antichain in $\mathcal{D}$ has size $2^{\aleph_0}$.*

Exercise 9.2.34 *Show that in Exercise* treeofmin *9.2.32we may also guarantee that $G'' \equiv_T G \vee 0''$ for every $G \in [T]$.*

Exercise 9.2.35 *Show that for every $\mathbf{c} \geq \mathbf{0}''$ there is a minimal degree $\mathbf{g}$ with $\mathbf{g}'' = \mathbf{c} = \mathbf{g} \vee \mathbf{0}''$.*

Definition 9.2.36 *A binary tree T is* pointed *if every $A \in [T]$ computes T. It is* uniformly pointed *if there is an e such that $\Phi_e^A = T$ for every $A \in [T]$.*

Exercise 9.2.37 *Relativize Theorem* spmindeg *9.2.21 to an arbitrary degree $\mathbf{c}$ to prove that every degree $\mathbf{c}$ has a* minimal cover*, i.e. a $\mathbf{g} > \mathbf{c}$ such that the open interval $(\mathbf{c}, \mathbf{g})$ is empty. Hint: One can proceed as usual by adding a $C \in \mathbf{c}$ into all oracle computations or one can use uniformly pointed trees recursive in C. In this case, just use binary function trees recursive in C which are subtrees of the tree T defined by $T(\sigma)(2n) = C(n)$ and $T(\sigma)(2n+1) = \sigma(n)$.*

Exercise 9.2.38 *All of the other results of this section now relativize.*

Exercise 9.2.39 *Prove that every strictly ascending sequence of degrees has a minimal upper bound $\mathbf{g}$. Hint: If the given sequence is $\mathbf{c}_n$, use uniformly pointed trees of degree $\mathbf{c}_n$ for some n.*

Exercise 9.2.40 *Show that the* $\mathbf{g}$ *of the previous exercise can be constructed so that* $\mathbf{g}'' \leq \oplus \mathbf{c}_n''$.

Exercise 9.2.41 *Show that one can also get two minimal upper bounds* $\mathbf{g}_0$ *and* $\mathbf{g}_1$ *for the* $\mathbf{c}_n$ *of the previous exercise with* $(\mathbf{g}_0 \vee \mathbf{g}_1)'' \leq \oplus \mathbf{c}_n''$. *Note that these* $\mathbf{g}_i$ *form an exact pair for the ideal generated by the* $\mathbf{c}_n$.

Exercise 9.2.42 *Thus if in the previous two exercises* $\mathbf{c}_n = \mathbf{0}^{(n)}$ *then one gets a minimal upper bound* $\mathbf{g}$ *for the* $\mathbf{0}^{(n)}$ *such that* $\mathbf{g}'' = \mathbf{0}^{(\omega)}$ *and indeed two such (which then form an exact pair for the arithmetic degrees) with* $(\mathbf{g}_0 \oplus \mathbf{g}_1)'' = \mathbf{0}^{(\omega)}$.

Exercise 9.2.43 *Prove that there is a tree* T *such that each path on* T *is a minimal upper bound for the ascending sequence* $\mathbf{c}_n$.

Definition 9.2.44 *A tree* T *is a* delayed e-splitting tree *if for every* n *there is an* $m > n$ *such that the strings* $T(\sigma)$ *for* $|\sigma| = m$ *are pairwise* e*-splititng.*

Exercise 9.2.45 *Prove the computation lemma for delayed* e*-splitting trees.*

Exercise 9.2.46 *Uniform trees; strongly uniform* $= 1$*-trees. one every path of minimal degree,* $F : \mathbb{N} \to \{0, 1, 2\}$*. minimal degrees generate* $\mathcal{D}$ *minimal* m*-degree Perhaps write out??*

Exercise 9.2.47 *other applications??*

sacksmin

9.3 Partial trees and Sacks minimal degrees

Sacksmin

Theorem 9.3.1 (Sacks) *There is a minimal degree below* $0'$.

Our plan is to use partial recursive binary trees in a construction recursive in $0'$. We have already seen (Lemmas nosplits 9.2.16, complemma 9.2.19, nosplitsoresplit 9.2.20 and Proposition esplits 9.2.18 that we can handle both the diagonalization and minimality requirements by using subtrees of the form $Fu(T, \sigma)$ and $Sp(T, e)$ even if they are partial as long as we do not run into a node with no convergent extensions on the trees we are using. Now $0'$ can recognize this situation when it occurs. Thus the problem is what to do when we arrive at a node with no extensions on a tree. Of course, we must change the tree we intend our set to be on but we must do so in a way that eventually stabilizes so that, for each requirement, we remain, from some point onward, on some partial tree that satisfies the requirement.

Proof. At stage s, we have already specified an initial segment α_s of the set A of minimal degree that we are building and a sequence (of indices for) nested partial recursive trees $T_{0,s} \geq_S T_{1,s} \geq_S \cdots \geq_S T_{k_s,s}$ with α_s on each of them (i.e. there are $\sigma_{i,s}$ such that $T_{i,s}(\sigma_{i,s}) = \alpha_s$). In fact, $\alpha_s = T_{k,s}(\emptyset)$. $T_{0,0}$ is the identity function on binary strings. (Indeed, as will become clear, $T_{0,s}$ is the identity function for every s.) Each $T_{i+1,s}$ is either $Sp(Fu(T_{i,s}, j), i)$ for some $j \in \{0, 1\}$or $Fu(T_{i,s}, \sigma)$ for some σ and is devoted to satisfying the minimality requirement for Φ_i with the choice of j devoted to satisfying the diagonalization requirements.

We now find the least $i \leq k_s$ such that $T_{i,s}(\sigma_{i,s}\hat{\ }0) \uparrow$. Let $k_{s+1} = i$ if one such exists, and let $k_{s+1} = k_s + 1$ otherwise. Note that this can be done recursively in $0'$ as we have indices for each $T_{i,s}$ as a partial recursive function.

- In the first case, we know that $T_{i,s}(\sigma_{i,s}\hat{\ }0) \uparrow$ while $T_{i-1,s}(\sigma_{i-1,s}\hat{\ }0) \downarrow$. Note that in this case T_i obviously cannot be of the form $Fu(T_{i-1}, \sigma)$ and so (by the rules of the construction which we are maintaining by induction) must be of the form $Sp(Fu(T_{i,s}, j), i)$. Thus by Proposition 9.2.18 there are no extensions of α_s on T_{i-1} which i-split. We now let $T_{k_s+1} = Fu(T_{i-1,s}, \sigma_{i-1.s}\hat{\ }0)$ (with the intention of satisfying the minimality requirement for Φ_i by being on a tree with no i-splits).

- In the second case, we let $T_{k_s+1,s+1} = Sp(Fu(T_{k_s,s}, j), k_s)$ where we choose j so that $\Phi_{k_s} \neq T_{k_s+1,s+1}(\emptyset)$ (to be specific, say we choose $j = 1$ if $\exists x(T_{k_s,s}(1) \neq \Phi_{k_s}(x) \downarrow)$ and $j = 0$ otherwise) and with the hope that we remain on this tree and so satisfy the minimality requirement for Φ_{k_s} by being on a k_s-splitting tree.

- In either case, we let $T_{i,s+1} = T_{i,s}$ for $i < k_{s+1}$ and $\alpha_{s+1} = T_{k_{s+1},s+1}(\emptyset)$. The trees T_i are, of course, not defined at $s + 1$ for $i > k_{s+1}$.

We now claim that the $T_{i,s}$ stabilize, i.e. there is a tree $T_i = \lim_{s\to\infty} T_{i,s}$ and all the requirements are satisfied. Note that if $T_{i,s}$ reaches its limit by stage t then $k_s > i$ for $s > t$. Suppose, by induction, that $T_{i,s}$ first reaches its limit T_i at stage s. At $s+1$ we set $T_{i+1,s+1} = Sp(Fu(T_{i,s}, j), i)$ (for some j) and we satisfy the diagonalization requirement for Φ_i. If we never change $T_{i+1,t}$ at a $t > s$ then $T_{i+1} = Sp(Fu(T_{i,s}, j), i))$ and we satisfy the minimality requirement for Φ_i by Lemma 9.2.19. If there is a stage after s at which we first change T_{i+1}, i.e. $T_{i+1,t} \neq T_{i+1,t+1}$ it must be because we are in the first case at stage t and we set $k_{t+1} = i+1$ and $T_{i+1,t+1} = Fu(T_i, \sigma_{i,t}\hat{\ }0)$ because $Sp(Fu(T_{i,t}, j), i))(\sigma_{i+1,t}\hat{\ }0)$ is divergent. In this case, we can never change T_{i+1} again. (No smaller one ever changes by our choice of s and it can never be chosen as the least point of divergence as long as it is a full subtree of the previous tree.) Moreover, α_v remains on T_i on which there are no i-splits above α_t (Proposition 9.2.18). Thus we satisfy the minimality requirement for Φ_i by Lemma 9.2.16. ■

Note that, in contrast to the Spector minimal degrees, no set recursive in $0'$ (and so even those of minimal degree) is **0**-dominated by Theorem 8.2.3. In ?? we actually need to know a bit more about the set A of minimal degree that we have just constructed.

minwtt0' **Corollary 9.3.2** *The set A of minimal degree constructed above is actually $\leq_{wtt} 0'$.*

Proof. To see that $A \leq_{wtt} 0'$ we need a recursive function f such that $f(n)$ bounds use from $0'$ needed to compute $A(n)$. An abstract view of the above construction is that at each stage s we have a number $k_s \leq s + 1$ and a sequence of indices for partial trees $T_{i,s}$

for $i \le k_s$. (Note that $\alpha_s = T_{k_s,s}(\emptyset)$.) We then ask for each $i \le k_s$ if $T_{i,s}(\sigma_{i,s}\hat{\ }0)\downarrow$ where this question is equivalent to the one that asks if $\exists\tau(T_{i,s}(\tau) = T_{k_s,s}(\emptyset)\ \&\ T_{i,s}(\tau\hat{\ }0)\downarrow)$. Each possible set of answers to these questions determines $0 < k_{s+1} \le k_s + 1 \le s + 1$ and the indices for the $T_{i,s+1}$ for $i \le k_{s+1}$ except when they say that $k_{s+1} = k_s + 1$. In this case, we need to ask one more question of $0'$: is there an x such that $T_{k_s,s}(1) \ne \Phi_{k_s}(x)\downarrow$? Thus we can recursively lay out all possible routes of the construction as a tree which at level s is (at most) $s + 1$ branching along with the (at most $s + 1$ many) questions of $0'$ needed to determine at each node of the tree at level s what stage $s+1$ of the construction would be if the given node corresponds to the actual stage s of the construction. Now to compute $A(n)$ note that we extend α_s at every stage of the construction so we only need a recursive bound on the questions asked in any possible run of the construction for n many stages. As the indices for all the possible $T_{i,s}$ are uniformly computable from the various assumed answers at the previous stages, it is clear that there is a recursive bound on the questions that are needed in all possible runs of the construction for n many steps.
■

cksMinCover **Exercise 9.3.3** *Theorem* Sacksmin *9.3.1 and Corollary* minwtt0' *9.3.2 above relativize to arbitrary degrees* $\mathbf{c}$ *to give a minimal cover* $\mathbf{g}$ *of* $\mathbf{c}$ *with* $\mathbf{g} \le_{wtt} \mathbf{c}'$. *Moreover, for any* C *there is a* G *which is uniformly* ω*-r.e. in* C *such that* $\mathbf{g}$ *is a minimal cover of* $\mathbf{c}$. *(Recall Definition* omegare *4.3.12 and Exercise* wtt0' *4.3.15.) ?? do out for later?? Also this earlier exercise??*

Exercise 9.3.4 *??Show that for every* $\mathbf{d} > \mathbf{0}$ *there is a minimal degree* $\mathbf{g} \le_T \mathbf{d} \vee \mathbf{0}'$ *such that* $\mathbf{g} \not\le_{\mathbf{T}} \mathbf{d}$. *Hint my construction in L p. 192?? only for* $\mathbf{d} < \mathbf{0}'$ *??otherwise below* $\mathbf{d}'$*??*

0'treeofmin **Exercise 9.3.5** *Construct a tree* $T \le_T 0'$ *such that every path on* T *is of minimal degree.*

Cone avoiding?? join ?? Complementation??

9.4 Minimal degrees below degrees in $\mathbf{H}_1$ and $\mathbf{GH}_1$

We want to prove that if $\mathbf{h} \in \mathbf{GH}_1$ then there is a minimal degree $\mathbf{a} < \mathbf{h}$. The proof builds on the construction of a Sacks minimal degree with highness giving us an approximation to but is unusual in that it relies on the recursion theorem to make the approximations work.

Remark: Not below every $\mathbf{H}_2$ Lerman [??]).

Question 9.4.1 *If* $A >_T 0$ *is r.e. then there is a minimal degree below* A *[??]. Can one construct such a degree with the techniques presented in this chapter and the previous one or some variation of them?*

cone avoiding, join, complementation results?

9.5 Jumps of minimal degrees

At the end of §9.2 we analyzed the possible double jumps of Spector minimal degrees. In this section we want to investigate the possible single jumps of arbitrary minimal degrees. Note first that every minimal degree is $\mathbf{GL}_2$ because every $\overline{\mathbf{GL}_2}$ degree has a 1-generic degree below it by Theorem 8.3.3. We show later ?? that there are minimal degrees in both $\mathbf{GL}_1$ and $\mathbf{GL}_2 - \mathbf{GL}_1$. Finally, we completely characterize the jumps of minimal degrees by giving a new proof due to Lempp, J. Miller S. Ng and L. Yu of Cooper's jump inversion theorem that every $\mathbf{c} \geq \mathbf{0}'$ is the jump of a minimal degree. The situation below $\mathbf{0}'$ is more complicated. While there are both $\mathbf{L}_1$ and $\mathbf{L}_2 - \mathbf{L}_1$ minimal degrees, not every degree $\mathbf{c}$ which is r.e. in and low over $0'$ is the jump of a minimal degree below $\mathbf{0}'$ (refs?? Shore noninversion theorem, Cooper).

9.5.1 Narrow trees and $\overline{\mathbf{GL}_1}$ minimal degrees

To produce a minimal degree not in $\mathbf{GL}_1$ we must combine a diagonalization of A' against $\Phi_e(A \oplus 0')$. The key idea here are the narrow subtrees $N(T)$.

Definition 9.5.1 *The* narrow subtree $N(T)$ *of a total tree* T *is defined by recursion.* $N(T)(\emptyset) = T(\emptyset)$*. If* $N(T)(\sigma) = T(\tau)$ *then* $N(T)(\sigma\hat{\ }i) = T(\tau\hat{\ }0\hat{\ }i)$*.*

Proposition 9.5.2 *If* T *is recursive so is* $N(T)$ *and an index for it can be found uniformly recursively in one for* T*. Of course, as with any recursive tree the question of whether* $A \in [N(T)]$ *is* Π_1 *in* A *and the index for* $N(T)$ *and so uniformly recursive in* A'*, i.e. there is a recursive* f *such that* $(\forall A)(A \in [N(T)] \Leftrightarrow f(n) \in A')$ *where* n *is any index for* $N(T)$*.*

Our plan is to use narrow subtrees to diagonalize. Intuitively we stay on some $N(T)$ with index i until we see that $\Phi_e^{\alpha_s \oplus 0'}(f(i)) \downarrow = 1$. At that point we make A go off $N(T)$ and so guarantee that $A' \neq \Phi_e^{A \oplus 0'}$. Formally we prove that diagonalization is dense.

nardense **Lemma 9.5.3** *The sets* $F_e = \{T | (\forall G \in [T]) \neg(\Phi_e^{A \oplus 0'} = A')\}$ *are dense in the Spector notion of forcing and there is a density function which is uniformly recursive in* $0'$ *on (the indices for) recursive trees..*

Proof. Let n be an index for T and consider $N(T) = S$. If there is a σ such that $\Phi_e^{S(\sigma) \oplus 0'}(f(n)) \downarrow = 1$ then the desired extension $\hat{T}$ of T is $Fu(T, \tau\hat{\ }1)$ where $T(\tau) = S(\sigma)$. The point here is that no $A \in [\hat{T}]$ is on $S = N(T)$ while $\Phi_e^{A \oplus 0'}(f(n)) \downarrow = 1$ for every $A \in \hat{T}$ and so $\Phi_e^{A \oplus 0'} \neq A'$. On the other hand, if there is no such σ then $N(T)$ is the desired extension of T as $f(n) \in A'$ for every $A \in [N(T)]$ while $\neg(\Phi_e^{A \oplus 0'}(f(n)) = 1)$ for every $A \in [N(T)]$ by our case assumption. It is clear that finding the desired extension of T is recursive in $0''$. ∎

Theorem 9.5.4 *There is a minimal degree $\mathbf{g} \leq \mathbf{0}''$ with $\mathbf{g} \notin \mathbf{GL}_1$. We may also guarantee that $\mathbf{g}'' = \mathbf{0}''$.*

Proof. Simply add the dense sets F_e to the ones D_e and C_e in the proof of Theorem spmindeg 9.2.21 to be met in the construction of G. To guarantee that $\mathbf{g}'' = \mathbf{0}''$ add in the dense B_e of Proposition totdense 9.2.24. ■

Exercise 9.5.5 *Modify the proof of Theorem sacksmin 9.3 to construct an $A \leq_T 0'$ of minimal degree with degree not in $\mathbf{L}_1$. Hint: intersperse stages at which one puts $T_{i+1,s+1} = N(T_{i,s})$ and then stays in this tree until $\Phi_e^{\alpha_s}(f(n)) \downarrow = 1$ where e and n are as in Lemma hardense 9.5.3 for $T_{i,s}$.*

9.5.2 Cooper's jump inversion theorem

We want to prove that every degree $\mathbf{c} \geq \mathbf{0}'$ is the jump of a minimal degree. To do this we modify the definition of the e-splitting subtree in an attempt to force the jump when we can.

ejumpspl **Definition 9.5.6** *The e-**jump splitting subtree of** T, $JSp(T,e) = S$ is defined by recursion. $S(\emptyset) = T(\emptyset)$ which is labeled ω. Suppose $S(\sigma) = T(\tau)$ is defined and is labeled some $m \leq \omega$. We search simultaneously for $\tau_0, \tau_1 \supseteq \tau$ such that $T(\tau_0)|_e T(\tau_1)$ and for a $\rho \supseteq \tau$ and an $n < m$ such that $\Phi_n^{T(\rho)}(n) \downarrow$ but $\Phi_n^{T(\tau)}(n) \uparrow$. If we first (in some canonical search order) find an e-split then we let $S(\sigma\hat{\ }i) = T(\tau_i)$ and label them both ω. If we first find a ρ and n as described we let $S(\sigma\hat{\ }0) = T(\rho)$ and label it n. $S(\sigma\hat{\ }1)$ is undefined in this case. (Of course, if neither search terminates, $S(\sigma\hat{\ }i) \uparrow$ for both $i = 0, 1$.)*

Proposition 9.5.7 *If T is (partial) recursive then so is $JSp(T,e)$ and an index for it can be found uniformly recursively in one for T.*

noisol **Lemma 9.5.8** *If $JSp(T,e) = S$, then there are no isolated paths on S, i.e. if $A \in [S]$ then there are infinitely many σ such that $S(\sigma) \subset A$ and $S(\sigma\hat{\ }i) \downarrow$ for $i = 0, 1$.*

Proof. This is immediate from the fact that whenever $S(\sigma) \downarrow$ but not both of $S(\sigma\hat{\ }i)$ are defined then only $S(\sigma\hat{\ }0)$ is defined and its label is in $\mathbb{N}$ and remains strictly decreasing until we reach a $\sigma\hat{\ }0^t$ such that both $S(\sigma\hat{\ }0^t\hat{\ }0)$ and $S(\sigma\hat{\ }0^t\hat{\ }1)$ are defined and their labels are ω. Thus we can continue to extend only one side (necessarily the 0 one) as we follow A on S only finitely often. ■

jcomplemma **Lemma 9.5.9** *If $S = JSp(T,e)$, $G \in [S]$ and Φ_e^G is total then $G \leq_T \Phi_e^G$.*

Proof. As for the basic Computation Lemma complemma 9.2.19, we compute an ascending sequence γ_n of initial segments of G (and so G itself) from Φ_e^G by recursion. We also compute σ_n and τ_n such that $T(\tau_n) = S(\sigma_n) = \gamma_n$ and its label m_n on S. We begin with $\gamma_0 = T(\emptyset) = S(\emptyset)$ which is an initial segment of G since $G \in [S]$. Suppose we have

$\gamma_n = S(\sigma_n) = T(\tau_n) \subset G$ and m_n. As $G \in [S]$, one of $S(\sigma_n\hat{\ }0)$ and $S(\sigma_n\hat{\ }1)$ is also an initial segment of G. We follow the procedure given in the definition of $JSp(T,e)(\sigma_n\hat{\ }i)$. If we first find an e-split then both $S(\sigma_n\hat{\ }i)$ are convergent. As they e-split we can decide which one is an initial segment of G using Φ_e^G as in the basic Computation Lemma and continue our recursion. If instead, we first find a new convergence for $\Phi_{\hat{n}}^{T(\rho)}(\hat{n})$ for $\hat{n} < n$, only $S(\sigma_n\hat{\ }0)$ is defined and it is then the next initial segment γ_{n+1} of G as required. Of course, $\sigma_{n+1} = \sigma_n\hat{\ }0$. This also supplies us with the next τ_{n+1} and $m_{n+1} = \hat{n}$. ■

lowmin **Theorem 9.5.10** *There is an A of minimal degree with $A' \equiv_T 0'$.*

Proof. The construction is similar to that for Theorem 9.3.1 except that we use e-jump splitting subtrees instead of e-splitting subtrees and we have to be a bit more careful about how we go off the partial trees.

At stage s, we have an already specified initial segment α_s of A and a sequence (of indices for) nested partial recursive trees $T_{0,s} \geq_S T_{1,s} \geq_S \cdots \geq_S T_{k_s,s}$ with α_s on each of them, indeed with $\alpha_s = T_{k,s}(\emptyset)$. $T_{0,s}$ is the identity function for every s. Each $T_{i+1,s}$ is either $JSp(Fu(T_{i,s},\sigma),i)$ for some σ or $Fu(T_{i,s},\sigma)$ for some σ.

We begin our search for k_{s+1} with $T_{k_s,s}$. We ask if $T_{k_s,s}(1)\downarrow$. If it is, so is $T_{k_s,s}(0)$. We then set $T_{k_{s+1}} = JSP(Fu(T_{k_s},j),k_s)$ where we choose j so that $\Phi_{k_s}(x) \neq T_{k_s,s}(j)(x)$ for some x and set $k_{s+1} = k_s + 1$. If $T_{k_s,s}(1)\uparrow$ we ask if $T_{k_s,s}(0)\downarrow$. If so we repeat our procedure with T_{k_s} replaced by $Fu(T_{k_s},0)$. By Lemma 9.5.8 this process eventually terminates either with an m such that $T_{k_s}(0^m\hat{\ }1)\downarrow = Fu(T_{k_s},0^m)(1)\downarrow$ and so a definition of $k_{s+1} = k_s + 1$ and $T_{k_{s+1}} = JSP(Fu(T_{k_s},0^m\hat{\ }j),k_s)$ or an m such that $T_{k_s}(0^m)\downarrow$ but $T_{k_s}(0^{m+1})\uparrow$ (m could be 0 and we take $0^0 = \emptyset$). In the later case, we move to T_{k_s-1} beginning with the σ_1 such that $T_{k_s-1}(\sigma_1) = T_{k_s}(0^m)$ and asking if $T_{k_s-1}(\sigma_1\hat{\ }1)\downarrow$. Continuing in this way we eventually reach l and m such that $T_{l,s}(\sigma\hat{\ }0^m\hat{\ }j)\downarrow$ for some σ and each $j \in \{0,1\}$ as $T_{0,s}$ is always the identity function and so defined at $\sigma\hat{\ }1$ for every σ. We now let $k_{s+1} = l + 1$ and $T_{k_{s+1}} = Fu(T_{l,s},\sigma\hat{\ }0^{m+1})$. We conclude the stage by setting $\alpha_{s+1} = T_{k_{s+1}}(\emptyset)$. Note that we extend α_s at every stage and $A = \cup\alpha_s \leq_T 0'$.

It is clear that the construction and so A is recursive in $0'$. We must now verify that the $T_{i,s}$ stabilize to trees T_i, all the requirements to make A of minimal degree and that $A' \leq_T 0'$. We argue much as in the proof of Theorem 9.3.1 for the first two claims:

Note again that if $T_{i,s}$ reaches its limit by stage t then $k_s > i$ for $s > t$. Suppose, by induction, that $T_{i,s}$ first reaches its limit T_i at stage s. At $s+1$ we set $T_{i+1,s+1} = JSP(Fu(T_{i_s},0^m\hat{\ }j),i)$ for some m and j as the only other possibilities change T_i. This action satisfies the diagonalization requirement for Φ_i. If we never change $T_{i+1,t}$ at a $t > s$ then $T_{i+1} = JSP(Fu(T_i,0^m\hat{\ }j),i))$ and we satisfy the minimality requirement for Φ_i by Lemma 9.5.9. If there is a first stage after s at which we change T_{i+1}, i.e. $T_{i+1,t} \neq T_{i+1,t+1}$, then it must be that we reached a situation with $T_{l,t}(\sigma\hat{\ }0^{\hat{m}}\hat{\ }\hat{\jmath})\downarrow$ for some σ and both $\hat{\jmath} \in \{0,1\}$ with l the first such we find in our search starting with k_t and moving downward and $\hat{m}$ the least such for l. As we now redefine T_{l+1} it must be that $l = i$ by our induction hypothesis. As t is the first stage after s at which we change T_{i+1},

$T_{i+1,t} = JSP(Fu(T_{i,s}, 0^m\hat{\ }j), i)$. As we did not end our search for this l with $l+1 = i+1$, if $T_{l,t}(\sigma) = T_{i+1,t}(\tau)$ then $T_{i+1,t}(\tau\hat{\ }0)\uparrow$. By the definition of $T_{i+1,t} = JSP(Fu(T_{i,s}, 0^m\hat{\ }j), i)$ this means that there are no i-splits on $T_{i,s} = T_i$ above σ. As $A \in [Fu(T_i, \sigma)]$ we satisfy the minimality requirement for Φ_i by Lemma nosplits 9.2.16. Once T_{i+1} is a full subtree of T_i (as it is at $t+1$), it can never be changed again as that would change some T_k for $k \leq i$ contrary to our choice of $s < t$.

To compute A' from $0'$ find a stage of the construction s at which we end the construction with $l \leq k_s$ and $T_{l,s}(\sigma\hat{\ }0^m\hat{\ }j)\downarrow$ for $j \in \{0,1\}$ and we let $k_{s+1} = l+1$ and $T_{l+1,s+1} = Fu(T_{l,s}, \sigma\hat{\ }0^{m+1})$. In this case we have $T_{l+1,s}(\tau\hat{\ }0)\uparrow$ where $T_{l+1,s}(\tau) = T_{l,s}(\sigma)$. If n is the label of $T_{l+1,s}(\tau)$, this means that there is no extension ρ of $T_{l+1,s}(\tau)$ on $T_{l,s}$ such that $\Phi_{\hat{n}}^{\rho}(\hat{n})\downarrow$ but $\Phi_{\hat{n}}^{T_{l+1,s}(\tau)}(\hat{n})\uparrow$ for $\hat{n} < n$. We now claim that, for $\hat{n} < n$, $\hat{n} \in A \Leftrightarrow \Phi_{\hat{n}}^{T_{l+1,s}(\tau)}(\hat{n})\downarrow$. As long as α_t stays on $T_{l,s}$ for $t > s$ (as it is now) the claim is obvious. The only way α_t can leave $T_{l,s}$ for the first time after s at t is for the same situation to occur with $l_1 < l$. In this case, the associated label must be $n_1 \geq n$ (as no new convergences below n can occur as long as we remain on $T_{l,s}$). In this case, no new convergences below n_1 can occur as long as we remain on $T_{l_1,t}$. This process must halt and so we eventually stay on some tree $T_{\hat{l},\hat{t}}$ on which there are no new convergences below some $\hat{n} \geq n$. To see that our original search in this procedure must find such stages s with arbitrarily large n, fix an r and start with a stage u by which $\forall e \leq r(\Phi_e^A(e)\downarrow \Leftrightarrow \Phi_e^{\alpha_u}(e)\downarrow)$. Now consider a $v > u$ for which Φ_v is the empty function. When we reach the first stage w at which $k_w = v+1$ for the first time after T_i has reached its limit for $i \leq v$ we set $T_{v+1,w+1} = JSP(Fu(T_{v,w}, 0^m\hat{\ }j), v))$ for some m and j. This tree has $T_{v+1,w+1}(\tau\hat{\ }1)\uparrow$ for every τ and so we would act as described above and for an $n \geq r$. ■

Theorem 9.5.11 *There is a binary function tree $T \leq_T 0'$ such that every $A \in [T]$ is of minimal degree and, moreover, $A' \equiv_T A \vee 0'$.*

Proof. We define T by recursion beginning with $T(\emptyset) = \emptyset$. Along each path in T we are using the construction of Theorem lowmin 9.5.10 with the change that when we would have chosen one $j \in \{0,1\}$ and set $T_{k_s+1} = Fu(S, j)$ for some S we follow both possibilities and define the next branching in T as the result of the two choices of j in the original construction. Thus at any node ρ when we have $T(\rho)$ defined we have an associated run of the above construction during which we have chosen $j = \rho(m)$ at the mth instance where we had to choose a j in the construction. To define $T(\rho\hat{\ }i)$ we now continue the construction as in the previous theorem until we reach the next stage s at which we must choose a j and set $T_{k_s+1} = JSp(Fu(T_{k_s,s}, j), k_s)$. We now let $T(\rho\hat{\ }j) = JSp(Fu(T_{k_s,s}, j), k_s)(\emptyset)$ for $j \in \{0,1\}$ and associate the version of the above construction in which we choose j with $T(\rho\hat{\ }j)$. ■

Corollary 9.5.12 (Cooper's Jump Inversion Theorem) *For every* $\mathbf{c} \geq \mathbf{0}'$ *there is a minimal degree* $\mathbf{a}$ *such that* $\mathbf{a}' = \mathbf{c} = \mathbf{a} \vee \mathbf{0}'$.

Proof. Take $C \in \mathbf{c}$ and let $A = \cup T(C \restriction n)$. ■

remark not all degrees REA in $\mathbf{0}'$ and low over it are jumps of minimal degrees below $\mathbf{0}'$ references.

Theorem lowmin 9.5.10 originally by Yates showed minimal below every nonrecursive r.e. degree and was already known (Theorem ??) that there are low nonrecursive r.e. degrees. Then ...

9.6 The minimal degrees generate $\mathcal{D}$

Our goal in this section is to prove that the minimal degrees generate $\mathcal{D}$ under join and meet. More specifically we prove that for every $\mathbf{a}$ there are minimal degrees $\mathbf{m}_0$, $\mathbf{m}_1$, $\mathbf{m}_2$ and $\mathbf{m}_3$ such that $\mathbf{a} = (\ \mathbf{m}_0 \vee \mathbf{m}_1) \wedge (\mathbf{m}_2 \vee\ \mathbf{m}_3)$. Our forcing conditions in this section are all recursive binary trees but we need a yet more restricted notion of tree. We begin with uniform trees (which play a crucial role in the next chapter) and strongly uniform trees or 1-trees.

1treedef **Definition 9.6.1** *A binary tree T is* uniform *if for every n there are $\rho_{n.0}, \rho_{n,1} \in 2^{<\omega}$ such that $T(\sigma\hat{\ }i) = T(\sigma)\hat{\ }\rho_{n,i}$ for every σ of length n. T is* strongly uniform *if, in addition, for every n, $\rho_{n.0}$ and $\rho_{n,1}$ are* adjacent, *i.e. there is exactly one j such that $\rho_{n.0}(j) \neq \rho_{n,1}(j)$. Strongly uniform trees are also called* 1-trees.

In this section all trees are recursive 1-*trees* and they are the conditions in our basic notion of forcing $\mathcal{P}$ with the usual notion of subtree as the extension relation. As $Fu(T, \sigma)$ is clearly a 1-tree for any 1-tree T, the diagonalization requirements D_e of Lemma spdiag 9.2.12 are still dense so we can meet those conditions as usual. Lemma nopslits ?? applies to any binary tree and so if our generic filter includes a tree with no e-splits then again, if Φ_e^G is total it is recursive. The computation lemma (complemma 9.2.19) also applies quite generally and so if the sets C_e of Lemma nosplitsoresplit 9.2.20 are dense then any generic for forcing with 1-trees is also of minimal degree. Thus we must show that if the 1-tree T has no extensions without e-splits then it has an extension which is e-splitting. It is actually helpful in this setting to first provide the analog of $Tot(T, e)$ that forces totality and proves the density of the B_e of Lemma totdense 9.2.24.

1totdense **Lemma 9.6.2** *The sets $B_e = \{T | \exists x \forall \sigma (\Phi_e^{T(\sigma)}(x) \uparrow)$ or $(\forall \sigma)(\forall x < |\sigma|)(\Phi_e^{T(\sigma)}(x) \downarrow)\}$ are dense in $\mathcal{P}$.*

Proof. Given a 1-tree T we define a partial recursive function $S = Tot_1(T, e)$ by recursion beginning as usual with $S(\emptyset) = T(\emptyset)$. Let $\{\sigma_i | i < 2^n\}$ list all the strings of length n and assume that $S(\sigma_i) = T(\tau_i)$ has been defined for all $i < 2^n$. To define S for all ρ of length $n+1$, we search first for a μ_0 such that $\Phi_e^{T(\tau_0\hat{\ }\mu_0)}(n) \downarrow$. Then we recursively search for μ_i such that $\Phi_e^{T(\tau_0\hat{\ }\mu_0\hat{\ }\dots\hat{\ }\mu_i)}(n) \downarrow$. If we eventually find μ_i for all $i < 2^n$, then we let $\mu = \mu_0\hat{\ }\dots\hat{\ }\mu_{2^n-1}$ and set $S(\sigma_i\hat{\ }j) = T(\tau_i\hat{\ }\mu\hat{\ }j)$ for $j \in \{0,1\}$. As T is a 1-tree it is easy to see that, if total, so is $Tot_1(T, e)$ and it satisfies the second clause of B_e. If it is not total then there is some n, τ_i and ν such that $T(\tau_i\hat{}\mu)\uparrow$ for every $\mu \supseteq \nu$. In this case, $Fu(T,\tau_i\hat{}\nu)$ satisfies the first clause of B_e with $x = n$. ■

We can now prove the remaining lemma that shows that all (sufficiently) generic G for $\mathcal{P}$ are of minimal degree.

itsoresplit **Lemma 9.6.3** *The sets $C_e = \{T|\ \exists x\forall\sigma(\Phi_e^{T(\sigma)}(x)\uparrow)$ or there are no e-splits on T or T is an e-splititng 1-tree$\}$ are dense in $\mathcal{P}$.*

Proof. By Lemma 1totdense 9.6.2, we may assume that the second clause of B_e is satisfied by T, i.e. $(\forall\sigma)(\forall x < |\sigma|)(\Phi_e^{T(\sigma)}(x)\downarrow$. We may also assume that there is no extension of T that satisfies the second clause of C_e so we can find e-splits on any $R \subseteq T$. We now wish to define an e-splitting 1-tree $Sp_1(T,e) = S \subseteq T$. We begin with converting arbitrary e-splits into ones that are adjacent and then defining two new operations on 1-trees.

Claim 9.6.4 *For any $R \subseteq T$ there are adjacent σ and τ such that $R(\sigma)|_e R(\tau)$ and so, in particular, for any ρ there are $\sigma,\tau \supseteq \rho$ which are adjacent such that $R(\sigma)|_e R(\tau)$.*

Proof. By our second assumption on T there are μ and ν such that $R(\mu)|_e R(\nu)$. Without loss of generality we may take $|\mu| = |\nu| > n$ where $R(\mu)$ and $R(\nu)$ e-split at n. Consider then the sequence $\langle\sigma_i|i \le k\rangle$ of adjacent binary strings of length n such that $\sigma_0 = \mu$ and $\sigma_k = \nu$. By our first assumption on $T \supseteq R$, $\Phi_e^{R(\sigma_i)}(n)\downarrow$ for every $i \le k$. As the first and last of these have different values there must be an i such that $\Phi_e^{R(\sigma_i)}(n)\downarrow \ne \Phi_e^{R(\sigma_{i+1})}(n)\downarrow$. Our desired adjacent e-split is then given by $\sigma = \sigma_i$ and $\tau = \sigma_{i+1}$. ■

Definition 9.6.5 *For any tree R and $\mu \in 2^{<\omega}$ we define R^μ (the* transfer tree *of R over μ) for $|\nu| \le |R(\emptyset)|$ as the tree such that, for every $\sigma \in 2$, $R^\mu(\sigma)$ is the string gotten from $R(\sigma)$ by replacing its initial segment of length $|\mu|$ by μ. For $R \subseteq T$ we define a new type of subtree $S = Sp_0(R,e)$. We begin by using the above Claim to construct sequences σ_i^0 and σ_i^1 for $i \in \mathbb{N}$ with σ_i^0 and σ_i^1 adjacent such that first, $R(\sigma_0^0)|_e R(\sigma_0^1)$ and then, in general, $R(\sigma_0^0\hat{}\cdots\hat{}\sigma_i^0\hat{}\sigma_{i+1}^0)|_e R(\sigma_0^0\hat{}\cdots\hat{}\sigma_i^0\hat{}\sigma_{i+1}^1)$. We now define S by recursion with $S(\emptyset) = R(\emptyset)$ and $S(\rho) = R(\Sigma\sigma_i^{\rho(i)})$ (where we use summation notation Σ for concatenation and the number of terms concatenated is $|\rho|$).*

Remark 9.6.6 *Note that as R is a 1-tree and the σ_i^0 and σ_i^1 are adjacent, S is also a 1-tree and, of course, $S \subseteq R$. Moreover, $S(\sigma)|_e S(\tau)$ for any $\sigma \ne \tau$ as the strings extend some e-split $R(\sigma_0^0)|_e R(\sigma_0^1)$ or $R(\sigma_0^0\hat{}\cdots\hat{}\sigma_i^0\hat{}\sigma_{i+1}^0)|_e R(\sigma_0^0\hat{}\cdots\hat{}\sigma_i^0\hat{}\sigma_{i+1}^1)$ for $i \ge 0$.*

■

Proof continued. We now define our e-splitting 1-tree $Sp_1(T,e) = S \subseteq T$ by recursion beginning with $S(\emptyset) = T(\emptyset)$. Let $\{\sigma_i|i < 2^n\}$ list all the strings of length n and assume that $S(\sigma_i) = T(\tau_i)$ has been defined for all $i < 2^n$. We let $R_0 = Sp_0(T_{\tau_0},e)$ and for $0 < i < 2^n$ we let $R_i = Sp_0(R_{i-1}^{T(\tau_i)},e)$ and $R = R_{2^n-1}$. We now let $S(\sigma_i\hat{}j) = R^{T(\tau_i)}(j)$. The verifications that this defines the next level of an e-splitting 1-tree contained in T

are straightforward. By the definition of Sp_0, $|R(\emptyset)| = |R_i(\emptyset)| = |T(\tau_i)|$ for every $i < 2^n$ and $R(0)$ and $R(1)$ are adjacent. By the definition of the transfer trees, $R^{T(\tau_i)}(0)$ and $R^{T(\tau_i)}(1)$ are adjacent extensions of $T(\tau_i) = S(\sigma_i)$ and the extensions are given by the same pair of strings for each i as R is a 1-tree. Moreover, since $R \subseteq R_i$ for every $i < 2^n$, each $R^{T(\tau_i)}(j)$ is a node on $R_i = Sp_0(R_{i-1}^{\tau_i}, e)$ (where $R_{-1} = T$) and so by the Remark above, they form an e-splitting.

This completes the definition of level $n+1$ of S and so, by recursion of $S = Sp_1(T, e)$ which is an e-splitting 1-tree extending T as required to establish the density of the C_e. ■

We have now shown the forcing with 1-trees produces a minimal degree.

1treemin **Proposition 9.6.7** *Any generic meeting the dense sets B_e, C_e and D_e for forcing with 1-trees is of minimal degree.*

Exercise 9.6.8 *Show that there are generics G as in Proposition 9.6.7 with $G \leq_T 0''$ and indeed with $G'' \equiv_T 0''$.*

Exercise 9.6.9 *Show that the generic sets of Proposition 9.6.7 are of minimal m-degree.*

We next want a tree of such minimal degrees, i.e. a 1-tree T such that every path is of minimal degree. We move to a tree of trees as we did in Exercise 9.2.32.

1treeofmin **Theorem 9.6.10** *If we force with the notion of forcing $\mathcal{P}_{1t}$ with conditions (T, n) where T is a 1-tree, $n \in \mathbb{N}$ and extension is defined by $(S, m) \leq_{\mathcal{P}_t} (T, n)$ if $S \leq_{\mathcal{S}} T$, $m \geq n$ and $S(\sigma) = T(\sigma)$ for every σ of length $\leq n$ and $V((T, n))$ is the finite binary 1-tree given by restricting T to strings of length n, then any sufficiently generic G is a 1-tree such that every path on G is of minimal degree.*

Proof. It is clear that G is a 1-tree from the fact that $V(P)$ is a 1-tree for every condition P and that if $Q \leq_{\mathcal{P}_1} P$ then $V(Q) \supseteq V(P)$ as 1-trees. To prove the theorem it suffices, by the last Proposition, to show that for each of the dense sets B_e, C_e and D_e any condition (T, n) there is a condition $(R, n) \leq_{\mathcal{P}_1} (T, n)$ such that for each σ of length n, $R_{R(\sigma)}$ is in the desired dense set. List the strings of length n as σ_i, $i < 2^n$. Begin with $S_0 = T_{T(\sigma_0)}$. By the relevant Lemma above (9.6.2, **??** and 9.2.12) we can refine S_0 to a 1-tree S_1 which is in the dense set. We can then consider $S_1^{T(\sigma_1)}$ and refine it to S_2 which is also in the dense set. We continue in this way to define S_i for $i < 2^n$ by refining $S_i^{T(\sigma_i)}$ to get an S_{i+1} in the dense set. At the end we have $S = S_{2^n}$ such that $S^{T(\sigma_i)}$ is in the dense set for each $i < 2^n$. We now define R by $R(\rho) = T(\rho)$ for $|\rho| \leq n$ and for $\rho \supset \sigma_i$ $R(\rho) = S^{T(\sigma_i)}(\rho)$. It is clear that $(R, n) \leq_{\mathcal{P}_1} (T, n)$ and for each σ of length n, $R_{R(\sigma)}$ is in the desired dense set. Let G be a generic 1-tree meeting all these dense sets. Now any $M \in [G]$ is $\mathcal{P}$-generic for the previous notion of forcing with 1-trees to the extent required by Proposition 9.6.7 and so is of minimal degree by that Proposition. ■

Exercise 9.6.11 *Show that there are generics G as in Theorem* 1treeofmin *9.6.10 with $G \leq_T 0''$ and indeed with $G'' \equiv_T 0''$.*

Exercise 9.6.12 *For each $n \geq 3$, Show that there are sets A of minimal degrees which are Σ_n but not Δ^0_n. Hint: take a path in the G of Theorem* 1treeofmin *9.6.10 which follows a path $C \in \Sigma_n - \Delta^0_n$, i.e. $A = \cup\{G(C \restriction k) | k \in \mathbb{N}\}$. (For $n = 2$, the result can be proven using, among other things, Exercise* 0'treeofmin *9.3.5.)??*

Finally, we use $\mathcal{P}_t$ to prove our main theorem for this section.

mingen **Theorem 9.6.13** *For every degree* $\mathbf{a}$ *there are minimal degrees* $\mathbf{m}_0$, $\mathbf{m}_1$, $\mathbf{m}_2$ *and* $\mathbf{m}_3$ *such that* $\mathbf{a} = (\ \mathbf{m}_0 \vee \mathbf{m}_1) \wedge (\mathbf{m}_2 \vee\ \mathbf{m}_3)$.

Proof. For any 1-tree G and set C, we let d_n be the unique x such that $G(\sigma\hat{\ }0)(x) \neq G(\sigma\hat{\ }1)(x)$ for any σ of length n and G^C be the path through G such that $G^C(d_n) = C(n)$. (As G is a 1-tree the x as required to define d_n is unique for each σ and the same for all of them.) These notions apply to finite 1-trees as G and finite binary strings as C with the obvious comment that there may only be finitely many d_n involved. If G is sufficiently generic for $\mathcal{P}_{1t}$ as required for Theorem 1treeofmin 9.6.10, and A is any set then it is clear that $A \leq_T G^A \vee G^{\bar{A}}$ as $n \in A$ if and only if $G^A(x) = 1$ where x is the nth place at which G^A and $G^{\bar{A}}$ differ (it is actually d_n). Thus we have two minimal degrees which join above $\mathbf{a}$. Our plan now is to take G_0 and G_1 two mutually sufficiently generic 1-trees for $\mathcal{P}_{1t}$ where the notion sufficiently generic now depends on A and assures that $(G_0^A \vee G_0^{\bar{A}}) \wedge (G_1^A \vee G_1^{\bar{A}})$.

Formally we consider the notion of forcing $\mathcal{P}_{2t}$ whose conditions consist of pairs $(P.Q)$ with each of P and Q a condition in P_{1t}. The ordering is given by $(\hat{P}.\hat{Q}) \leq_{\mathcal{P}_{2t}} (P.Q)$ if $\hat{P} \leq_{\mathcal{P}_{1t}} P$ and $\hat{Q} \leq_{\mathcal{P}_{1t}} Q$. In addition to the dense sets defined by requiring that each coordinate get into the dense sets from Theorem 1treeofmin 9.6.10 we have one more family of dense sets for the new meet requirement. For $(T, n) = P \in \mathcal{P}_{1t}$ we let P_n be the finite 1-tree given by restricting T to strings of length at most n. The argument is by now familiar. We let $A_e = \{(P.Q) | \exists x(\Phi_e^{(P_n^A \vee P_n^{\bar{A}})}(x) \downarrow \neq \Phi_e^{(Q_n^A \vee Q_n^{\bar{A}})}(x) \downarrow)$ or $(\forall(\hat{P}.\hat{Q}) \leq_{\mathcal{P}_{2t}} (P.Q))(\neg\exists x(\Phi_e^{(P_n^A \vee P_n^{\bar{A}})}(x) \downarrow \neq \Phi_e^{(Q_n^A \vee Q_n^{\bar{A}})}(x) \downarrow))\}$. Now if our generic meets A_e in condition $((P, n), (Q, m))$ and the first clause holds then clearly $\Phi_e^{(G_0^A \vee G_0^{\bar{A}})}(x) \downarrow \neq \Phi_e^{(G_1^A \vee G_1^{\bar{A}})}(x) \downarrow$ as, by the definition of extension in P_{2t}, $P_n^A \vee P_n^{\bar{A}}$ and $Q_n^A \vee Q_n^{\bar{A}}$ are initial segments of $G_0^A \vee G_0^{\bar{A}}$ and $G_1^A \vee G_1^{\bar{A}}$, respectively. On the other hand, if the second clause holds and $\Phi_e^{(G_0^A \vee G_0^{\bar{A}})}$ and $\Phi_e^{(G_1^A \vee G_1^{\bar{A}})}$ are total and equal, then we claim they are recursive in A. To compute $\Phi_e^{(G_0^A \vee G_0^{\bar{A}})}(x)$ find any finite extension R_m of P_n to a 1-tree of height m that is a subtree of P and such that $\Phi_e^{(R_n^A \vee R_n^{\bar{A}})}(x) \downarrow$. This R_m then extends to a full 1-tree R such that $(R, n) \leq_{\mathcal{P}_{1t}} (P, n)$. There must be one as $P_n \subseteq G_0$ and the computation of $\Phi_e^{(G_0^A \vee G_0^{\bar{A}})}(x)$ only requires finitely many levels of $G_0 \subseteq P$. If this were not the correct answer then, as for P and G_0, there would be a finite extension S_k of Q_n contained in Q which gives the same answer as $\Phi_e^{(G_1^A \vee G_1^{\bar{A}})}(x) \downarrow$. Again this S_k can be extended to an S such that

$(S,n) \leq_{\mathcal{P}_{1t}} (Q,n)$. Then $((R,n),(S,n)) \leq_{\mathcal{P}_{2t}} (P,Q)$ but satisfies the first clause of A_e for the desired contradiction. ■

Exercise 9.6.14 *Show that the minimal degrees in* $\overline{\mathbf{GL}}_1$ *generate* $\mathcal{D}$.

Chapter 10

initialseg

Lattice Initial Segments of $\mathcal{D}$

Known results, history. Plan and goals here. Include all finite lattices and all countable distributive ones two of the major steps in previous process. new proof based on ... sufficient for all Applications. do two quantifier theory decidable and three undecidable.

First present the proof for recursive lattices which suffices for all our applications. Then indicate how to extend argument to cover all sublattices of any recursive lattice and so, for example, all distributive lattices.

??Explain intuition for construction: combine lattice tables as used to embed lattices and tree constructions used to build minimal degrees. What problems arise when try to merge and how adjust forcing conditions to overcome them. some done below do more. ??

10.1 Lattice Tables, trees and the notion of forcing

Our plan is to use lattice tables like those of latrep 6.3.8 to provide the basics of our embeddings of lattices as initial segments of $\mathcal{D}$. For simple embeddings in §latembsec 6.3 we used a Cohen like forcing with conditions that were finite sequences of elements of our representation. In light of our move to trees in §spectormin 9.2 to construct minimal degrees, it should not be surprising that we now move to conditions that are trees built on lattice tables Θ, i.e. maps $T : \Theta^{<\omega} \to \Theta^{<\omega}$ to provide the appropriate notions of forcing. The generic G that is built is then, as in §latembsec 6.3, an infinite sequence of elements from Θ. As a first approximation, the embedding is given as before. For $x \in \mathcal{L}$, $x \longmapsto G_x$ where $G_x(n) = G(n)(x)$. (Recall that the elements of Θ are maps from $\mathcal{L}$ to $\mathbb{N}$.) Order, nonorder, join and meet are handled much as in §latembsec 6.3. The key idea for making the embedding onto an initial segment again uses a type of e-splitting tree. While we want to deal with infinite lattices, a crucial component of the computation lemma for e-splitting trees (even in the minimal degree case) is that the trees are finitely branching. As long as they are finitely branching, one has a hope of determining the path taken by using Φ_e^G to choose among the e-splits. Thus we approximate our table by finite subsets Θ_i and consider trees T that at level i branch according to the elements of Θ_i. We also have an associated decomposition of our given lattice $\mathcal{L} = \cup\mathcal{L}_i$. Now if one ignores the meet operation and the required interpolants it is easy to get a finite lattice table for a finite lattice. We call these upper semilattice (usl) tables. We postpone the meet interpolants for $\mathcal{L}_i$ to Θ_{i+1}. While this is not strictly necessary, it makes the construction of the tables much easier. Moreover, we need a new type of interpolant to make the embeddings constructed be onto initial segments of $\mathcal{D}$ and we do not know if these could be incorporated as well into a finite lattice table for $\mathcal{L}_i$.

These new interpolants are called homogeneity interpolants. The idea here is that if we intend to force $\Phi_e^G \equiv_T G_x$ for some $x \in \mathcal{L}$ then using G_x we cannot distinguish among all the nodes σ in the tree at a given level n as many have the same σ_x (be congruent modulo x). This suggests that we want our trees to have some kind of homogeneity guaranteeing that what happens above one such σ is congruent to what happens above any other $\tau \equiv_x \sigma$. Of course, we need this property for every $x \in \mathcal{L}$.

With the above as a brief motivation, we now formally define the lattice tables that we use and the associated trees.

uslrepdef **Definition 10.1.1** *Let Θ be a set of maps from an usl $\mathcal{L}$ with least element 0 and greatest element 1 into $\mathbb{N}$. For $\alpha, \beta \in \Theta$ and $x \in \mathcal{L}$, we write $\alpha \equiv_x \beta$ (α is congruent to β modulo x) if $\alpha(x) = \beta(x)$. We write $\alpha \equiv_{x,y} \beta$ to indicate that α is congruent to β modulo both x and y. Such a Θ is an* usl table for $\mathcal{L}$ *if it contains the function that is 0 on every input (which we, by an abuse of notation, denote by 0) and for every $\alpha, \beta \in \Theta$ and $x, y, z \in \mathcal{L}$ the following properties hold:*

1. *$\alpha(0) = 0$.*
2. *(Differentiation) If $x \nleq y$ then there are $\gamma, \delta \in \Theta$ such that $\gamma \equiv_y \delta$ but $\gamma \not\equiv_x \delta$.*
3. *(Order) If $x \leq y$ and $\alpha \equiv_y \beta$ then $\alpha \equiv_x \beta$.*
4. *(Join) If $x \vee y = z$ and $\alpha \equiv_{x,y} \beta$ then $\alpha \equiv_z \beta$.*

restriction **Notation 10.1.2** *If Θ is an usl representation for $\mathcal{L}$ and $\hat{\mathcal{L}} \subseteq \mathcal{L}$ then we denote the* restriction *of Θ to $\hat{\mathcal{L}}$ by $\Theta \restriction \hat{\mathcal{L}} = \{\alpha \restriction \hat{\mathcal{L}} | \alpha \in \Theta\}$. We also say that Θ is an* extension *of $\Theta \restriction \hat{\mathcal{L}}$. Note that as all our (upper or lower semi)lattices contain 1, the order property guarantees that if $\alpha \restriction \hat{\mathcal{L}} = \beta \restriction \hat{\mathcal{L}}$ then $\alpha = \beta$. Thus when we extend an usl representation $\hat{\Theta}$ for $\hat{\mathcal{L}}$ to one Θ for $\mathcal{L}$ (as in the constructions for Proposition extend 10.3.4 we can use the same $\alpha \in \hat{\Theta}$ to denote its unique extension in Θ.*

homo **Definition 10.1.3** *If Θ' and Θ are usl representations for $\mathcal{L}'$ and $\mathcal{L}$, respectively, $\hat{\mathcal{L}} \subseteq \mathcal{L}' \subseteq \mathcal{L}$ and $f : \Theta' \to \Theta$, then f is an $\hat{\mathcal{L}}$-*homomorphism *if, for all $\alpha, \beta \in \Theta'$ and $x \in \hat{\mathcal{L}}$, $\alpha \equiv_x \beta \Rightarrow f(\alpha) \equiv_x f(\beta)$.*

Theorem 10.1.4 (see Theorem repthm 10.3.1) *If $\mathcal{L}$ is a countable lattice, then there is an usl table Θ for $\mathcal{L}$ along with a nested sequence of finite sublower semilattices,* slsls, *$\mathcal{L}_i$ starting with $\mathcal{L}_0 = \{0, 1\}$ with union $\mathcal{L}$ and a nested sequence of finite subsets Θ_i with union Θ with both sequences recursive in $\mathcal{L}$ with the following properties:*

1. For each i, $\Theta_i \restriction \mathcal{L}_i$ is an usl table for $\mathcal{L}_i$.

2. There are meet interpolants for Θ_i in Θ_{i+1}, i.e. if $\alpha \equiv_z \beta$, $x \wedge y = z$ (with $\alpha, \beta \in \Theta_i$ and $x, y, z \in \mathcal{L}_i$) then there are $\gamma_0, \gamma_1, \gamma_2 \in \Theta_{i+1}$ such that $\alpha \equiv_x \gamma_0 \equiv_y \gamma_1 \equiv_x \gamma_2 \equiv_y \beta$.

3. For every sublowersemilattice $\hat{\mathcal{L}}$ of $\mathcal{L}_i$, $\hat{\mathcal{L}} \subseteq_{lsl} \mathcal{L}_i$, there are homogeneity interpolants for Θ_i with respect to $\hat{\mathcal{L}}$ in Θ_{i+1}, i.e. for every $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \Theta_i$ such that $\forall w \in \hat{\mathcal{L}}(\alpha_0 \equiv_w \alpha_1 \rightarrow \beta_0 \equiv_w \beta_1)$, there are $\gamma_0, \gamma_1 \in \Theta_{i+1}$ and $\hat{\mathcal{L}}$-homomorphisms $f, g, h : \Theta_i \rightarrow \Theta_{i+1}$ such that $f : \alpha_0, \alpha_1 \mapsto \beta_0, \gamma_1$, $g : \alpha_0, \alpha_1 \mapsto \gamma_0, \gamma_1$ and $h : \alpha_0, \alpha_1 \mapsto \gamma_0, \beta_1$, i.e. $f(\alpha_0) = \beta_0$, $f(\alpha_1) = \gamma_1$ etc.

We prove this theorem in §lattablesec 10.3. Our goal in this and the next section is to prove that we have initial segment embeddings for all recursive lattices.

clatticeiso **Theorem 10.1.5** *Every recursive lattice $\mathcal{L}$ is isomorphic to an initial segment of $\mathcal{D}$.*

For the rest of this section and all of the next we fix a recursive lattice $\mathcal{L}$ and a sequence $\langle \mathcal{L}_i, \Theta_i \rangle$ for it as specified in Theorem repthm 10.3.1. We now move on to the definition of the trees that are the conditions in our forcing relation.

latticetree **Definition 10.1.6** *A* tree T (for the sequence $\langle \mathcal{L}_i, \Theta_i \rangle$), *which we call simply a* tree *in this chapter ,is a recursive function such that for some $k \in \omega$ its domain is the empty string $\emptyset$ and all strings in the Cartesian product $\prod_{n=0}^{n=m} \Theta_{k+n}$ for each $m \in \omega$. We denote this number k by $k(T)$. For each $\sigma \in \operatorname{dom} T$, $T(\sigma) \in \prod_{n=0}^{n=q} \Theta_n$ for some $q \geq |\sigma| - 1$. Moreover, T has the following properties for all $\sigma, \tau \in \operatorname{dom} T$:*

1. *(Order) $\sigma \subseteq \tau \Rightarrow T(\sigma) \subseteq T(\tau)$.*

2. *(Nonorder) $\sigma | \tau \Rightarrow T(\sigma) | T(\tau)$. In fact, we specifically require that, for every $\sigma \in \prod_{n=0}^{n=m} \Theta_{k+n}$ and $\alpha \in \Theta_{k+m+1}$, $T(\sigma\hat{\ }\alpha) \supseteq T(\sigma)\hat{\ }\alpha$.*

3. *(Uniformity) For every fixed length l there is, for each $\alpha \in \Theta_{k+l}$, a string $\rho_{l,\alpha}$ so that, for a given l, all the $\rho_{l,\alpha}$ are of the same length independently of α and if $|\sigma| = l$ then $T(\sigma\hat{\ }\alpha) = T(\sigma)\hat{\ }\rho_{l,\alpha}$. Note that by the nonorder property (2), for fixed l and $\alpha \neq \beta$, $\rho_{l,\alpha} \neq \rho_{l,\beta}$, in fact, by our specific requirement, $\rho_{l,\alpha}(0) = \alpha$.*

Thus our trees T have branchings of width $|\Theta_{k(T)+n}|$ at level n and satisfy order and nonorder properties as for Spector forcing. In addition, they enjoy a strong uniformity property that plays a crucial role in our verifications.

subtree **Definition 10.1.7** *We say that a tree S is a* subtree *of a tree T, $S \subseteq T$, if $k(S) \geq k(T)$ and $(\forall \sigma \in \operatorname{dom} S)(\exists \tau \in \operatorname{dom} T)[S(\sigma) = T(\tau)]$.*

We note three useful facts that illuminate the structure of subtrees. The first says that the branchings on S follow those on T.

Lemma 10.1.8 *If S is a subtree of T then*
1. $S(\sigma) = T(\tau) \to (\forall\alpha \in \Theta_{k(S)+|\sigma|})(S(\sigma\hat{}\alpha) \supseteq T(\tau\hat{}\alpha)$
2. $\forall l \exists \rho_{l,\alpha} \forall \sigma, \tau(|\sigma| = l \ \& \ S(\sigma) = T(\tau) \ \to \ S(\sigma\hat{}\alpha) = T(\tau\hat{}\rho_{l,\alpha})$ for $\alpha \in \Theta_{k(S)+l}$ and
3. $[S] \subseteq [T]$.

Proof. The first fact follows immediately from our specific implementation of the nonorder property for trees (Definition latticetree 10.1.6(2)). The second follows from the uniformity requirements (3) of Definition latticetree 10.1.6 for S and T as well as property (2). the last is immediate from the definition. ■

Transitivity of the subtree relation should be clear but an even stronger claim is proven in Proposition trans 10.1.11. We mention some specific operations on trees that we use later.

fulldef **Definition 10.1.9** *If T is a tree and $\sigma \in \operatorname{dom} T$ then $Fu(T, \sigma)$ or T_σ is defined by $T_\sigma(\tau) = T(\sigma\hat{}\tau)$. Clearly, $k(T_\sigma) = k(T) + |\sigma|$ and $T_\sigma \subseteq T$. Note that for $\sigma \in \operatorname{dom} T$ and $\tau \in \operatorname{dom} T_\sigma$, $(T_\sigma)_\tau = T_{\sigma\hat{}\tau}$. For a string $\mu \in \prod_{n=0}^{n=q} \Theta_n$ with $q \leq |T(\emptyset)| - 1$, we let T^μ (the* transfer tree *of T over μ) be the tree such that, for every $\sigma \in \operatorname{dom} T$, $T^\mu(\sigma)$ is the string gotten from $T(\sigma)$ by replacing its initial segment of length $q + 1$ (which is contained in $T(\emptyset)$) by μ. We write T^μ_σ for $(T_\sigma)^\mu$. Finally, if T is a tree with $k(T) = k$ and $\sigma \in \operatorname{dom} T$ then we let $T^*_\sigma = T_\sigma \upharpoonright \operatorname{dom} T$. Clearly, $k(T^*_\sigma) = k(T)$ and $T^*_\sigma \subseteq T$. Note that for $\sigma \in \operatorname{dom} T$ and $\tau \in \operatorname{dom} T^*_\sigma$, $(T^*_\sigma)^*_\tau = T^*_{\sigma\hat{}\tau}$.*

A crucial notion for our constructions is that of preserving the congruences of specified slsls of our given lattice $\mathcal{L}$.

prescong **Definition 10.1.10** *If $\hat{\mathcal{L}}$ is a finite slsl of $\mathcal{L}$ we say that a subtree S of T* preserves the congruences *of $\hat{\mathcal{L}}$, $S \subseteq_{\hat{\mathcal{L}}} T$, if $\hat{\mathcal{L}} \subseteq \mathcal{L}_{k(T)}$ and, whenever $x \in \hat{\mathcal{L}}$, $S(\sigma) = T(\tau)$, $\alpha \equiv_x \beta$, $S(\sigma\hat{}\alpha) = T(\tau\hat{}\mu)$ and $S(\sigma\hat{}\beta) = T(\tau\hat{}\nu)$, then $\mu \equiv_x \nu$. Here α and β are members of the appropriate Θ_i and μ and ν are sequences (necessarily of the same length m) of elements from the appropriate Θ_j's. We say that such sequences μ and ν are congruent modulo x, $\mu \equiv_x \nu$, if $\mu(j) \equiv_x \nu(j)$ for each $j < m$.*

trans **Proposition 10.1.11** *If $R \subseteq_{\mathcal{L}_1} S \subseteq_{\mathcal{L}_2} T$ and then $R \subseteq_{\mathcal{L}_1 \cap \mathcal{L}_2} T$.*

Proof. To see that $R \subseteq T$ note first that $k(R) \geq k(S) \geq k(T)$. Next suppose that $\rho \in \operatorname{dom} R$ and $\alpha \in \Theta_{k(R)+|\rho|}$. As $R \subseteq S$ we have a σ such that $R(\rho) = S(\sigma)$ and $R(\rho\hat{}\alpha) \supseteq S(\sigma\hat{}\alpha)$. As $S \subseteq T$ we have a τ such that $S(\sigma) = T(\tau)$ and $S(\sigma\hat{}\alpha) \supseteq T(\tau\hat{}\alpha)$. Thus $R(\rho) = T(\tau)$ and $R(\rho\hat{}\alpha) \supseteq T(\tau\hat{}\alpha)$ as required. As for the preservation of $\mathcal{L}_1 \cap \mathcal{L}_2$ congruences, suppose $R(\rho) = S(\sigma) = T(\tau)$, $x \in \mathcal{L}_1 \cap \mathcal{L}_2$, $\alpha_0, \alpha_1 \in \Theta_{k(R)+|\rho|}$ and $\alpha_0 \equiv_x \alpha_1$. Let $R(\rho\hat{}\alpha_i) = S(\sigma\hat{}\mu_i) = T(\tau\hat{}\nu_i)$. As $x \in \mathcal{L}_1$ and $R \subseteq_{\mathcal{L}_1} S$, $\mu_0 \equiv_x \mu_1$. As $x \in \mathcal{L}_2$ and

$S \subseteq_{\mathcal{L}_2} T$ it then follows by induction on the (by uniformity, necessarily common) length of μ_i that $\nu_0 \equiv_x \nu_1$ as required.

The details of this induction follow. Write $\nu_i = \nu_i^0\hat{}\cdots\hat{}\nu_i^s$ where $S(\sigma\hat{}\mu_i(0)\cdots\hat{}\mu_i(t)) = T(\tau\hat{}\nu_i^{0\hat{}\cdots\hat{}}\nu_i^t)$. Then inductively $\mu_0(j) \equiv_x \mu_1(j)$ gives $\nu_0^j \equiv_x \nu_1^j$. For $j = 0$ this follows directly from Definition 10.1.10. For the inductive step, consider, without loss of generality, the case $j = 1$. We have $S(\sigma\hat{}\mu_0(0)\hat{}\mu_0(1)) = T(\tau\hat{}\nu_0^0\hat{}\nu_0^1)$ and $S(\sigma\hat{}\mu_1(0)\hat{}\mu_1(1)) = T(\tau\hat{}\nu_1^0\hat{}\nu_1^1)$. Consider $S(\sigma\hat{}\mu_0(0)\hat{}\mu_1(1)) = T(\tau\hat{}\nu_0^0\hat{}\nu)$ for some ν. By the uniformity clause (3) of Definition 10.1.6, there is a ζ such that $S(\sigma\hat{}\mu_0(0)\hat{}\mu_1(1)) = S(\sigma\hat{}\mu_0(0))\hat{}\zeta$ and $S(\sigma\hat{}\mu_1(0)\hat{}\mu_1(1)) = S(\sigma\hat{}\mu_1(0))\hat{}\zeta$. Thus $T(\tau\hat{}\nu_0^0\hat{}\nu) = T(\tau\hat{}\nu_0^0)\hat{}\zeta$ and $T(\tau\hat{}\nu_1^0\hat{}\nu_1^1) = T(\tau\hat{}\nu_1^0)\hat{}\zeta$. Again, by the uniformity clause and the uniqueness of the $\rho_{l,\alpha}$ there (iterated $|\nu|$ times), $\nu = \nu_1^1$. Finally, by Definition 10.1.10 again, $\nu \equiv_x \nu_0^1$ as $\mu_1(1) \equiv_x \mu_0(1)$ and so $\nu_1^1 \equiv_x \nu_0^1$ as required. ■

We now present the notion of forcing for constructing our embedding of $\mathcal{L}$ as an initial segment of $\mathcal{D}$.

defforcing **Definition 10.1.12** *The* forcing conditions P *our notion of forcing* $\mathcal{P}$ *are trees* T *(for* $\langle \mathcal{L}_i, \Theta_i \rangle$*). We say* $T_1 \leq_{\mathcal{P}} T_0$*,if* $T_1 \subseteq_{K(T_0)} T_0$ *where, as often, we denote* $\mathcal{L}_{k(T)}$ *by* $K(T)$*. We let* $V(T) = T(\emptyset)$*. The top element of* $\mathcal{P}$ *consists of the identity tree* Id *(which has* $k(Id) = 0$*).*

fulllem **Lemma 10.1.13** *If* T *is a tree,* $\sigma \in \operatorname{dom} T$ *and* $\hat{\mathcal{L}} \subseteq \mathcal{L}_{k(T)}$*, then* $T_\sigma \subseteq_{\hat{\mathcal{L}}} T$*. If* $\sigma, \tau \in \operatorname{dom} T$ *are of the same length and* $S \leq_{\mathcal{P}} T_\sigma$ *then* $S^{T(\tau)} \leq_{\mathcal{P}} T_\tau$*. We also have that* $T_\sigma^{T(\tau)} = T_\tau$*.*

Proof. The first assertions follow directly from the definitions. The last two follow from the uniformity assumption on our trees. ■

It is easy to see that sets $C_n = \{P |\, |V(P)| > n \;\&\; k(P) > n\}$ are dense. Just extend to some P_σ. We assume that any generic filter $\mathcal{G}$ we consider meets these sets. It then determines a generic function $G \in \prod_{n=0}^{\infty} \Theta_n$, i.e. a function on ω with $G(n) \in \Theta_n$. On this basis we could naively try to define our embedding of $\mathcal{K}$ into $\mathcal{D}$ as we did in §6.3: For $x \in \mathcal{K} \subseteq \mathcal{L}$ we let $G_x : \omega \to \omega$ be defined by $G_x(n) = G(n)(x)$. The desired image of x would then be $\deg(G_x)$. Now the order and join properties of usl representations guarantee that this embedding preserves order and join (on all of $\mathcal{L}$ even). If $x \leq y$ then by the order property we can (recursively in the table $\langle \Theta_i \rangle$) calculate $G_x(m)$ from $G_y(m)$ by finding any $\alpha \in \Theta_m$ with $\alpha(y) = G_y(m)$ and declaring that $G_x(m) = \alpha(x)$. (Such an α exists since $G(m)$ is one.) Similarly if $x \vee y = z$ then, by the join property, we can calculate $G_z(m)$ from $G_x(m)$ and $G_y(m)$ by finding any $\alpha \in \Theta_m$ such that $\alpha(x) = G_x(m)$ and $\alpha(y) = G_y(m)$ and declaring that $G_z(m) = \alpha(z)$. (Again $G(m)$ is such an α.)

Were congruences modulo x always preserved for every x, we could directly carry out the diagonalization and other requirements as well for this definition of G_x. In actuality, however, not all congruences are preserved as we refine to various subtrees in our construction. Thus we must modify the definition of the images in $\mathcal{D}$ and provide nice representations of the degree corresponding to x.

Definition 10.1.14 *If $\mathcal{G}$ is a generic filter meeting the dense sets C_n, G the corresponding generic element of $\prod_{n=0}^{\infty} \Theta_n$, $P \in \mathcal{G}$ and $x \in K(P)$ then G^P is the sequence $\langle \alpha_n | n \in \omega \rangle$ where $P(\langle \alpha_n | n < m \rangle) \subseteq G$ for every m. (Thus $\langle \alpha_n \rangle$ is the path that G follows in the domain of P. In particular, $G = G^{Id}$. It is obvious from the definitions that G is a path on (i.e. in the range of) Q for every $Q \in \mathcal{G}$.) We define $G_x^P(n)$ as $\alpha_n(x)$.*

The crucial point is that the degree of G_x^P does not depend on P once $x \in K(P)$.

Lemma 10.1.15 *If $x \in K(P), K(Q)$ for P, Q in a generic $\mathcal{G}$, then $G_x^P \equiv_T G_x^Q$.*

Proof. As there is an $R \leq_{\mathcal{P}} P, Q$ in $\mathcal{G}$ by the compatibility of all conditions in a generic filter, it suffices to consider the case that $Q \leq_{\mathcal{P}} P$. Let $G^P = \langle \alpha_n \rangle$ and $G^Q = \langle \beta_n \rangle$. By the definition of subtree there is for each n an $m(n)$ such that $Q(\langle \beta_s | s < n \rangle) = P(\langle \alpha_s | s < m(n) \rangle)$ and we can compute the function m recursively in the trees. (By the uniformity of the trees, there is, for each n, a unique $m(n)$ such that $|Q(\sigma)| = |P(\tau)|$ for every σ of length n and every τ of length $m(n)$.) Moreover, by our definition of subtree, $\beta_n = \alpha_{m(n)}$. Thus $G_x^Q(n) = \beta_n(x) = \alpha_{m(n)}(x) = G_x^P(m(n))$ and so $G_x^Q \leq_h G_x^P$. The other direction depends on the congruence preservations for x implied by $Q \subseteq_{K(P)} P$.

Suppose that we have, by recursion, determined $G_x^P(i) = \alpha_i(x)$ for $i \leq m(n)$. The next step followed by G in Q is $\beta_{n+1} = \alpha_{m(n)+1}$. It corresponds to the sequence $\langle \alpha_i | m(n) + 1 \leq i < m(n+1) \rangle$. The definition of $\subseteq_{K(P)}$ implies that $\langle \alpha_i(x) | m(n) + 1 \leq i < m(n+1) \rangle$ is uniquely determined by $\beta_{n+1}(x)$ to continue the recursion. ■

Thus given a generic $\mathcal{G}$ we can define a map from $\mathcal{L}$ into $\mathcal{D}$ by sending $x \in \mathcal{L}$ to $\deg(G_x^P)$ for any $P \in \mathcal{G}$ with $x \in K(P)$. Our proof plans above as in latembsec §6.3 for the preservation of order and join now work here as well simply by applying them to G^P on P (in place of G on Id) for any $P \in \mathcal{G}$ with $x, y, z \in K(P)$. Thus we only need to verify the preservation of nonorder and that our map is onto an initial segment of $\mathcal{D}$. (Note that meet is preserved once we know that the mapping is an order isomorphism of the lattice $\mathcal{L}$ onto an initial segment of $\mathcal{D}$ as meet is definable from order. Of course, this argument would apply to join as well but no new work is needed to note that join is preserved. It is also worth commenting that we use the join structure in the usl tables as well as the meet interpolants in the proof that the embedding is onto an initial segment of $\mathcal{D}$.)

10.2 Initial segment conditions

To assure that our embedding preserves nonorder we want to show, for any $x \nleq y$ in $\mathcal{K}$, condition P with $x, y \in K(P)$ and Φ_e, that there is a $Q \leq_{\mathcal{P}} P$ such that for any $G \in [Q]$ $\Phi_e^{G_y^P} \neq G_x^P$ and a $Q \leq_{\mathcal{P}} P$ and $x \in K(Q)$ such that for any $G \in [Q]$ for which Φ_e^G is total, $\Phi_e^G \equiv_T G_x^Q$. These two results would then finish the proof of our theorem. We begin with the analog of total subtrees of Definition totdef 9.2.22 and the corresponding dense sets that make our task simpler.

totldef **Definition 10.2.1** *Let T be a condition in $\mathcal{P}$. If for every $\sigma \in \operatorname{dom} T$ and every x there is a $\tau \supseteq \sigma$ such that $\Phi_e^{T(\tau)}(x)\downarrow$ then we define a subtree $S = Totl(T,e)$ with the same domain by recursion on the length of $\sigma \in \operatorname{dom} T$. We begin with $S(\emptyset) = T(\emptyset)$. Suppose for every $\sigma_i \in \operatorname{dom} T$ of length n there is a τ_i such that we have defined $S(\sigma_i) = T(\tau_i)$ for $i < m$. We now list the α such that we must define $S(\rho_i\hat{}\alpha)$ to get the next level of S as $\langle \alpha_j | j < s\rangle$. We proceed to define ρ_l for each $l = \langle i, j\rangle$ with $i < m$ and $j < s$. (For convenience we assume these are the $l < r = m \cdot s$.) For $l = 0 = \langle 0, 0\rangle$ we search for the first $\rho \supseteq$ such that $\Phi_e^{T(\tau_0\hat{}\alpha_0\hat{}\rho)}(|\sigma|)\downarrow$. One exists by our assumption. We then set $\rho = \rho_0$. If we have defined ρ_l for $l \le q$ and $\mu_q = \rho_0\hat{}\ldots\hat{}\rho_q$ then we let ρ_{q+1} where $q+1 = \langle \hat{\imath}, \hat{\jmath}\rangle$ be the first ρ such that $\Phi_e^{T(\tau_{\hat{\imath}}\hat{}\alpha_{\hat{\jmath}}\hat{}\mu_q\hat{}\rho)}(|\sigma|)\downarrow$. We now let μ be the concatenation of the ρ_l for $l < r$ and set $S(\sigma_i\hat{}\alpha_j) = T(\tau_i\hat{}\alpha_j\hat{}\mu)$.*

Definition 10.2.2 *If T is a tree and $(\forall\sigma)(\forall x < |\sigma|)(\Phi_e^{T(\sigma)}(x)\downarrow$, we say that T is e-*total* and we denote $\Phi_e^{T(\sigma)}(n)$ for $n < |\sigma|$ by $q_T(n, \sigma)$.*

Lemma 10.2.3 *If T and $Totl(T) = S$ is defined then $S \le_{\mathcal{P}} T$ and S is e-*total*.*

Proof. The second claim is immediate from the definition of $Totl(T)$. As for the first, it is immediate that $\operatorname{dom} S = \operatorname{dom} T$ and that, since T is a tree, that S is a subtree of T. As the definition of S has $S(\sigma_i\hat{}\alpha_j) = T(\tau_i\hat{}\alpha_j\hat{}\mu)$ for a single μ over all the nodes $\sigma_i\hat{}\alpha_j$ of level $n+1$, it is also clear that S preserves all the congruences of $K(T)$. ■

totl **Lemma 10.2.4** *The sets $B_e = \{P | \exists x \forall \sigma (\Phi_e^{P(\sigma)}(x)\uparrow)$ or P is e-*total*$\}$ are dense in $\mathcal{P}$.*

Proof. Suppose we are given P and e. If there is an x and a σ such that $\Phi_e^{P(\tau)}(x)\uparrow$ for every $\tau \supseteq \sigma$, then clearly P_σ (or P^*_σ) satisfies the first clause in the definition of B_e. Otherwise, $Totl(P, e)$ satisfies the second clause by the last Lemma. ■

diag **Proposition 10.2.5 (Diagonalization)** *For any $x \nleq y$ in $\mathcal{L}$, $e \in \mathbb{N}$ and condition P with $x, y \in K(P)$, there is an $Q \le P$ such that $\forall G \in [Q]$, if Φ_e^G is total then $\Phi_e^{G_y^P} \ne G_x^P$.*

Proof. There is clearly an j such that $\Phi_e^{G_y^P} = \Phi_j^G$. By Lemma 10.2.4 we may assume that P is j-total. We then choose any $\alpha_0, \alpha_1 \in \Theta_{k(P)}$ such that $\alpha_0 \equiv_y \alpha_1$ but $\alpha_0 \not\equiv_x \alpha_1$. Such α_0 and α_1 exist by the differentiation property of usl tables. Let $\beta_i = (\alpha_i)^{|\sigma|}$, i.e. the concatenation of $|\sigma|$ many copies of α_i for $i \in \{0, 1\}$. Consider then the conditions P_{β_i} and any $G_i \in [P_{\beta_i}]$. Of course, $(G_i)_x^P(|\sigma|) = \alpha_i(x)$ while $\Phi_i^{P_{\beta_i}(\emptyset)}(|\sigma|)\downarrow$ for $i = 0, 1$ by Lemma 10.2.4. As the β_i, for $i = 0, 1$, are congruent modulo y and $y \in K(P)$, the initial segments of G_y^P that $P_{\beta_i}(\emptyset)$ determine are equal. Thus the $\Phi_i^{P_{\beta_i}(\emptyset)}(|\sigma|) = \Phi_e^{G_y^P}(|\sigma|)$ are convergent and equal. So for one $i \in \{0, 1\}$, $\Phi_i^{P_{\beta_i}(\emptyset)}(|\sigma|) \ne \alpha_i(x)$. For that i, P_{β_i} is the Q required in the Lemma. ■

We turn now to the requirement that the image of $\mathcal{K}$ under our embedding form an initial segment of $\mathcal{D}$. This argument is somewhat more complicated than those above and uses both the meet and homogeneity interpolants.

We begin with the notion of an e-splitting appropriate to our trees and a lemma about such splittings.

defsplit **Definition 10.2.6** *Given a Φ_e and an e-total tree Q. we say that σ and τ with $|\sigma| = |\tau|$ are an e-splitting or e-split on Q (modulo w) if ($\sigma \equiv_w \tau$ and) there is an $n < |\sigma|$ such that $q_T(n,\sigma) \neq q_T(n,\tau)$. If $R \leq Q, R(\mu) = Q(\sigma), R(\nu) = Q(\tau)$ and σ and τ e-split (modulo w) on Q then we also say that μ and ν e-split on R (modulo w).*

meetsplit **Lemma 10.2.7** *Given an e-total condition Q, there is a $\rho \in \operatorname{dom} Q$ such that the set $Sp(\rho) = \{w \in K(Q)|$ there are no σ, τ that e-split on Q^*_ρ modulo $w\}$ is maximal. Moreover, this maximal set is closed under meet and so has a least element z.*

Proof. Let $k = k(Q)$ and $\hat{\mathcal{K}} = K(Q)$. As $\hat{\mathcal{K}}$ is finite there is clearly a ρ such that $Sp(\rho)$ is maximal. Note that then $Sp(\mu) = Sp(\rho)$ for any $\mu \supseteq \rho$ with $\mu\backslash\rho \in \operatorname{dom} Q^*_\rho$ as $Q^*_\mu \subseteq_{\hat{\mathcal{K}}} Q^*_\rho$. Consider any $x, y \in Sp(\rho)$ with $x \wedge y = w$. As $\hat{\mathcal{K}} \subseteq_{lsl} \mathcal{L}$, $w \in \hat{\mathcal{K}}$. To show that $Sp(\rho)$ is closed under meet it suffices (by the maximality of $Sp(\rho)$) to show that there is no e-splitting on $Q^*_{\rho\hat{}0}$ modulo w. Remember that, by definition, $k = k(Q^*_{\rho\hat{}0}) = k(Q^*_\rho) = k(Q)$. Suppose there were such a split $\bar{\mu}$ and $\bar{\nu}$, each of length m. By our definition of $Q^*_{\rho\hat{}0}$, $\bar{\mu}, \bar{\nu} \in \prod_{n=0}^{n=m} \Theta_{k+n}$. In Q^*_ρ at the corresponding levels, however, there are branchings for all elements of Θ_{k+n+1}. (That is there are, for example, successors of $Q^*_\rho(0\hat{}\mu \restriction n+1)$ for every element of Θ_{k+n+1} while in $Q^*_\rho(\mu \restriction n+1)$ there are ones only for the elements of Θ_{k+n}.) Thus, by the existence of meet interpolants for Θ_{k+n} in Θ_{k+n+1}, there are $\bar{\gamma}_0, \bar{\gamma}_1, \bar{\gamma}_2 \in \prod_{n=0}^{n=m} \Theta_{k+n+1}$ such that for each $j \leq m$, the $\bar{\gamma}_i(j)$ for $i \in \{0,1,2\}$ are meet interpolants for $\bar{\mu}(j)$ and $\bar{\nu}(j)$, i.e. $\bar{\mu} \equiv_x \bar{\gamma}_0 \equiv_y \bar{\gamma}_1 \equiv_x \bar{\gamma}_2 \equiv_y \bar{\nu}$. As $\bar{\mu}$ and $\bar{\nu}$ form a e-splitting on $Q^*_{\rho\hat{}0}$ so do one of the successive pairs such as $0\hat{}\bar{\gamma}_0$, $0\hat{}\bar{\gamma}_1$ on Q^*_ρ. But then $0\hat{}\bar{\gamma}_0$ and $0\hat{}\bar{\gamma}_1$ would be an e-split on Q^*_ρ congruent modulo y for a contradiction. (The situations for the other pairs are the same but perhaps with x in place of y.) ■

We now build the analog of the e-splitting subtrees of Definition esplittree 9.2.17.

splittree **Proposition 10.2.8** *Given an e-total Q with $k(Q) = k$ and $K(Q) = \hat{\mathcal{K}}$ with ρ and z as in Lemma meetsplit 10.2.7, there is a condition $S \leq Q^*_\rho$ with $k(S) = k$ such that any $\sigma, \tau \in \operatorname{dom} S (= \operatorname{dom} Q)$ with $\sigma \not\equiv_z \tau$ e-split on S. (Of course, by the choice of ρ and z there are no e-splits on Q^*_ρ which are congruent modulo z.) Such a tree S is called a $z - e$*-splitting *tree.*

Proof. We define $S(\sigma)$ (with $k(S) = k$) by induction on $|\sigma|$ beginning, of course, with $S(\emptyset) = Q^*_\rho(\emptyset)$. Suppose we have defined $S(\sigma) = Q^*_\rho(\tau_\sigma)$ for all σ of length n. We must define $S(\sigma\hat{}\alpha)$ for all such σ and appropriate α as extensions $Q^*_\rho(\tau_{\sigma\hat{}\alpha})$ of $Q^*_\rho(\tau_\sigma\hat{}\alpha)$

obeying all the congruences in $\hat{\mathcal{K}}$, i.e. if $x \in \hat{\mathcal{K}}$ and $\alpha \equiv_x \beta$ then $\tau_{\sigma\hat{}\alpha} \equiv_x \tau_{\sigma\hat{}\beta}$. We list the σ of length $n+1$ as $\sigma_i\hat{}\alpha_i$ for $i < m = |\prod_{j=0}^{j=n} \Theta_{k+j}|$ and define by induction on $r < l = m(m+1)/2$ (the number of pairs $\{i,j\}$ with $i,j < m$) strings $\rho_{i,r}$ simultaneously for all $i < m$. At the end of our induction we set $\tau_{\sigma_i\hat{}\alpha_i} = \tau_{\sigma_i}\hat{}\alpha_i\hat{}\rho_{i,0}\hat{}\ldots\hat{}\rho_{i,l-1}$. For this to succeed it suffices to maintain uniformity and guarantee, for every $i,j < m$ and $w \in \hat{\mathcal{K}}$, that $\alpha_i \equiv_w \alpha_j \Rightarrow \rho_{i,r} \equiv_w \rho_{j,r}$ for every $r < l$ and that if $\alpha_i \not\equiv_z \alpha_j$ then $\tau_{\sigma_i}\hat{}\alpha_i\hat{}\rho_{i,0}\hat{}\ldots\hat{}\rho_{i,r}$ and $\tau_{\sigma_j}\hat{}\alpha_j\hat{}\rho_{j,0}\hat{}\ldots\hat{}\rho_{j,r}$ e-split on Q^*_ρ where $r < l$ is (the code for) $\{i,j\}$.

By induction on $r < l$ we suppose we have $\tau_{\sigma_i}\hat{}\rho_{i,0}\hat{}\ldots\hat{}\rho_{i,r-1} = \nu_i$ for all $i < m$ and that $\{p,q\}$ is pair number r. If $\alpha_p \equiv_z \alpha_q$ there is no requirement to satisfy and we let $\rho_{i,r} = \emptyset$ for every i. Otherwise, let w be the largest $y \in \mathcal{L}_{k+n}$ such that $\alpha_p \equiv_y \alpha_q$. (To see that there is a largest such y, first note that $\mathcal{L}_{k+n}$ is a lattice as it is a finite lsl. As Θ_{k+n} is an usl table for $\mathcal{L}_{k+n}$, if $\alpha_p \equiv_{u,v} \alpha_q$ for $u,v \in \mathcal{L}_{k+n}$ then $\alpha_p \equiv_t \alpha_q$ where t is the least element of $\mathcal{L}_{k+n}$ above both u and v (their join from the viewpoint of $\mathcal{L}_{k+n}$). Thus, there is a largest y as desired.) Of course, $z \not\leq w$. By our choice of z there are $\sigma, \tau \in \prod_{t=0}^{t=c} \Theta_{k+t}$ such that ν_p extended by σ and τ form an e-splitting congruent modulo w on Q^*_ρ. (We can find such a split on $Q^*_{\nu_p}$ by the definition of ρ and z and our assumption on w. It translates into such σ and τ.) Consider $\nu_q\hat{}\tau$. It must form an e-splitting on Q^*_ρ with one of $\nu_p\hat{}\sigma$ and $\nu_p\hat{}\tau$ by the basic properties of Q. If it splits with the latter string then we can set $\rho_{i,r+1} = \tau$ and clearly fulfill the requirements for this pair $\{p,q\}$ both for congruence modulo w (as all new extensions are identical) and e-splitting. Of course, uniformity is maintained as the $\rho_{i,r+1}$ are the same for all i. Thus we assume that $\nu_p\hat{}\sigma$ and $\nu_q\hat{}\tau$ e-split on Q^*_ρ. We now use our homogeneity interpolants.

We know that w is the largest $y \in \mathcal{L}_{n+k}$ such that $\alpha_p \equiv_y \alpha_q$ and that $\sigma \equiv_w \tau$. Thus for any $x \in \hat{\mathcal{K}} \subseteq \mathcal{L}_{k+n}$ if $\alpha_p \equiv_x \alpha_q$ then $x \leq w$ and so $\sigma \equiv_x \tau$. By Theorem repthm 10.3.1(3) we can now find homogeneity interpolants $\gamma_0(s), \gamma_1(s)$ in Θ_{k+s+1} and associated $\hat{\mathcal{K}}$-homomorphisms $f_s, g_s, h_s : \Theta_{k+s} \to \Theta_{k+s+1}$ such that $f_s : \alpha_p, \alpha_q \mapsto \sigma(s), \gamma_1(s)$, $g_s : \alpha_p, \alpha_q \mapsto \gamma_0(s), \gamma_1(s)$ and $h_s : \alpha_p, \alpha_q \mapsto \gamma_0(s), \tau(s)$ for each $s < |\sigma| = |\tau|$. (We let $\alpha_0 = \alpha_p$, $\alpha_1 = \alpha_q$, $\beta_0 = \sigma(s)$, $\beta_1 = \tau(s)$, $\hat{\mathcal{L}} = \hat{\mathcal{K}}$ and $i = k+s$ in the Theorem.) Note that the branchings in Q^*_ρ are at some levels up from the corresponding ones in $Q^*_{\nu_p}$ or $Q^*_{\nu_q}$ on which we chose σ and τ. Thus these homogeneity interpolants are available within the branchings in Q^*_ρ. As $\nu_p\hat{}\sigma$ and $\nu_q\hat{}\tau$ e-split on Q^*_ρ one of the pairs $\nu_p\hat{}\sigma, \nu_q\hat{}\bar{\gamma}_1$; $\nu_p\hat{}\bar{\gamma}_0, \nu_q\hat{}\bar{\gamma}_1$ and $\nu_p\hat{}\bar{\gamma}_0, \nu_q\hat{}\tau$ must also e-split on Q^*_ρ. Suppose, for the sake of definiteness, it is the second pair $\nu_p\hat{}\bar{\gamma}_0, \nu_q\hat{}\bar{\gamma}_1$. In this case, we let $\rho_{i,r+1}(s) = g_s(\alpha_i)$ for every i and s. Note that uniformity is maintained as $\rho_{i,r+1}(s)$ depends only on α_i. We use f_s or h_s in place of g_s if the e-splitting pairs are $\nu_p\hat{}\sigma, \nu_q\hat{}\bar{\gamma}_1$ or $\nu_p\hat{}\bar{\gamma}_0, \nu_q\hat{}\tau$, respectively. By the homomorphism properties of the interpolants these extensions preserve all the congruences in $\hat{\mathcal{K}}$ between any α_i and α_j as required to complete the induction and our construction of an e-splitting tree . ■

We now conclude the proof that our embedding maps onto an initial segment of $\mathcal{D}$. by showing that for $G \in [S]$ with S a $z-e$-splitting tree, $\Phi^G_e \equiv_T G^S_z$. The proof is analogous to that of the Computation Lemma (9.2.19).

comp **Lemma 10.2.9** *If S is a $z-e$-splitting tree and $G \in [S]$ then $\Phi_e^G \equiv_T G_z^S$.*

Proof. We first show that $\Phi_e^G \leq_T G_z^S$. Consider any n. Using G_z^S we can find all the $\sigma \in \operatorname{dom} S$ of length n such that $\sigma(l) = G_z^S(l)$ for every $l \leq n$. All of these σ are congruent modulo z and so all S_σ force the same value for Φ_e^G at n. As $S(\sigma)$ is an initial segment of G for one of these σ, this value must be $\Phi_e^G(n)$. We next argue that $G_z^S \leq_T \Phi_e^G$. Consider all $\sigma, \tau \in \operatorname{dom} S$ of length n. If $\sigma \not\equiv_z \tau$ then S_σ and S_τ force different values for Φ_e^G at some $l < n$. Thus using $\Phi_e^G \upharpoonright n$ we can find the unique congruence class modulo z consisting of those σ such that $S(\sigma)$ is not ruled out as a possible initial segment of G. For one σ in this class, $S(\sigma)$ is an initial segment of G and as all the σ in this class are congruent modulo z, they all determine the same values of $G_z^S \upharpoonright n$ which must then be the correct value. ■

We have now completed the proof that any generic filter $\mathcal{G}$ (deciding all sentences and meeting the dense sets provided by Lemma 10.2.4 and Propositions 10.2.5 and 10.2.8) provides an embedding of $\mathcal{L}$ onto an initial segment of $\mathcal{D}$ that sends x to $\deg(G_x^P)$ (for any $P \in \mathcal{G}$ with $x \in K(P)$). This establishes Theorem 10.1.5 given our lattice table theorem whose proof we provide in §10.3. We now indicate how to modify Theorem **??** so as to apply to any sublattice $\mathcal{K}$ of a recursive lattice.

clatticeiso **Theorem 10.2.10** *If $\mathcal{K}$ is a sublattice of a recursive lattice $\mathcal{L}$ then $\mathcal{K}$ is isomorphic to an initial segment of $\mathcal{D}$.*

Proof. The changes needed to the proof of Theorem 10.1.5 are mostly notational. The forcing conditions are now pairs $\langle T, \hat{\mathcal{K}} \rangle$ where T is a tree (for $\langle \mathcal{L}_i, \Theta_i \rangle$) and $\hat{\mathcal{K}}$ is a finite slsl of $\mathcal{K} \cap \mathcal{L}_{k(T)}$. We say that $\langle T_1, \mathcal{K}_1 \rangle \leq_{\mathcal{P}} \langle T_0, \mathcal{K}_0 \rangle$ if $T_1 \subseteq_{\mathcal{K}_0} T_0$ and $\mathcal{K}_1 \supseteq \mathcal{K}_0$. We let $V(\langle T, \hat{\mathcal{K}} \rangle) = T(\emptyset)$. If $P = \langle T, \hat{\mathcal{K}} \rangle$ is a condition we let $K(P) = \hat{\mathcal{K}}$, $Tr(P) = T$ and $k(P) = k(T)$. In following much of the original proof, one should often simply replace a condition P by $Tr(P)$ when $K(P)$ is fixed. Along these lines, for example, we use P_σ, P_σ^*, P^τ and P_σ^τ to stand for $\langle Tr(P)_\sigma, K(P) \rangle$, $\langle Tr(P)_\sigma^*, K(P) \rangle$, $\langle Tr(P)^\tau, K(P) \rangle$ and $\langle Tr(P)_\sigma^\tau, K(P) \rangle$, respectively. The top element of $\mathcal{P}$ consists of the identity tree Id (which has $k(Id) = 0$) and the slsl $\mathcal{L}_0 = \{0, 1\}$.

The basic dense sets C_n that we assume are met by any generic are now extended to include, for each $x \in \mathcal{K}$ the sets $\{P | x \in K(P)\}$. to see that these are dense, consider any $Q \in \mathcal{P}$ and $x \in \mathcal{K}$. Let $\mathcal{K}'$ be the slsl of $\mathcal{K}$ generated by $K(Q)$ and x and let $i \geq k(Q)$ be such that $\mathcal{K}' \subseteq \mathcal{L}_i$. Define S with $k(S) = i$ by $S(\sigma) = Tr(Q)(0^{i-k(Q)\hat{}}\sigma)$. Clearly, $\langle S, \mathcal{K}' \rangle \leq_{\mathcal{P}} Q$ and is in the required set.

The definition of the embedding, now from $\mathcal{K}$, into $\mathcal{D}$ is the same as before noting that $K(P)$ is now the second coordinate of P rather than simply $\mathcal{L}_{k(P)}$. The operations on trees and proofs used to verify the diagonalization (for $x, y \in \mathcal{K}$) and initial segment properties (with $\Phi_e^G = G_z^P$ for $z \in \mathcal{K}(\mathcal{Q}) \subseteq \mathcal{K}$) are now essentially the same. Just keep in mind that they are applied to $Tr(P)$ and $K(P)$ does not change. ■

This version of the theorem provides initial segment embeddings for many nonrecursive lattices. As an example we have the following corollary.

Corollary 10.2.11 *Every countable distributive lattice is isomorphic to an initial segment of $\mathcal{D}$.*

Proof. There is a recursive universal countable distributive lattice. In fact, every countable distributive lattice can be embedded into the atomless Boolean algebra.?? $\blacksquare$

Exercise 10.2.12 *Prove that the embedding of our recursive lattice $\mathcal{L}$ can be taken to be into $\mathcal{D}(\leq \mathbf{0}'')$ and, indeed that the generic G constructed has double jump $0''$. For embeddings of a sublattice $\mathcal{K}$ of $\mathcal{L}$ determine where the embedding lies and what can be said about G''.*

Exercise 10.2.13 *Prove that the embedding of our recursive lattice $\mathcal{L}$ is onto an initial segment of both the tt and wtt degrees. (Hint: recall Exercise* 0domtt *8.1.2.)*

Exercise 10.2.14 *Theorem* reclatticeiso *10.1.5 relativizes to any degree* $\mathbf{a}$ *and so every countable lattice $\mathcal{L}$ (with 0 and 1) is isomorphic to a segment of $\mathcal{D}$, i.e. to $[\mathbf{a},\mathbf{b}] = \{\mathbf{x}|\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}\}$ for some $\mathbf{b}$ where $\mathbf{a}$ is the degree of $\mathcal{L}$. Indeed, we may take $\mathbf{b}'' = \mathbf{a}''$.*

lattablesec

10.3 Constructing lattice tables

repthm **Theorem 10.3.1** *If $\mathcal{L}$ is a countable lattice then there is an usl table Θ of $\mathcal{L}$ along with a nested sequence of finite slsls $\mathcal{L}_i$ starting with $\mathcal{L}_0 = \{0,1\}$ with union $\mathcal{L}$ and a nested sequence of finite subsets Θ_i with union Θ with both sequences recursive in $\mathcal{L}$ with the following properties:*

1. *For each i, $\Theta_i \upharpoonright \mathcal{L}_i$ is an usl table of $\mathcal{L}_i$.*

2. *There are* meet interpolants *for Θ_i in Θ_{i+1}, i.e. if $\alpha \equiv_z \beta$, $x \wedge y = z$ (in Θ_i and $\mathcal{L}_i$, respectively) then there are $\gamma_0, \gamma_1, \gamma_2 \in \Theta_{i+1}$ such that $\alpha \equiv_x \gamma_0 \equiv_y \gamma_1 \equiv_x \gamma_2 \equiv_y \beta$.*

3. *For every $\hat{\mathcal{L}} \subseteq_{lsl} \mathcal{L}_i$ there are* homogeneity interpolants *for Θ_i with respect to $\hat{\mathcal{L}}$ in Θ_{i+1}, i.e. for every $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \Theta_i$ such that $\forall w \in \hat{\mathcal{L}}(\alpha_0 \equiv_w \alpha_1 \to \beta_0 \equiv_w \beta_1)$, there are $\gamma_0, \gamma_1 \in \Theta_{i+1}$ and $\hat{\mathcal{L}}$-homomorphisms $f, g, h : \Theta_i \to \Theta_{i+1}$ such that $f : \alpha_0, \alpha_1 \mapsto \beta_0, \gamma_1$, $g : \alpha_0, \alpha_1 \mapsto \gamma_0, \gamma_1$ and $h : \alpha_0, \alpha_1 \mapsto \gamma_0, \beta_1$.*

Proof. We first define the sequence $\mathcal{L}_i$ of slsls of $\mathcal{L}$ beginning with $\mathcal{L}_0$ which consists of the 0 and 1 of $\mathcal{L}$. We let the other elements of $\mathcal{L}$ be x_n for $n \geq 1$ and $\mathcal{L}_n$ be the (necessarily finite) slsl of $\mathcal{L}$ generated by $\{0, 1, x_1, \ldots, x_n\}$. As for Θ, we choose a countable set α_i and stipulate that $\Theta = \{\alpha_i | i \in \omega\}$. We begin defining the (values of) the α_i by setting $\alpha_0(x) = 0$ for all $x \in \mathcal{L}$ and $\alpha(0) = 0$ for all $\alpha \in \Theta$. We now define

Θ_n and the values of $\alpha \in \Theta_n$ (other than α_0) on the elements of $\mathcal{L}_n$ (other than 0) by recursion. For Θ_0 we choose a new element β of Θ and let $\Theta_0 = \{\alpha_0, \beta\}$ and set $\beta(1) = 1$. Given Θ_n and the values for its elements on $\mathcal{L}_n$ we wish to enlarge Θ_n to Θ_{n+1} and define the values of $\alpha(x)$ for $\alpha \in \Theta_{n+1}$ and $x \in \mathcal{L}_{n+1}$ so that the requirements of the Theorem are satisfied. To do this we prove a number of general extension theorems for usl representations in the Propositions below that show that we can make simple extensions to satisfy any particular meet or homogeneity requirement and also extend usl representations from smaller to larger slsls of $\mathcal{L}$. To be more specific, we first apply Proposition meetinterp 10.3.5 successively for each choice of $x \wedge y = z$ in $\mathcal{L}_n$ and $\alpha, \beta \in \Theta_n$ with $\alpha \equiv_z \beta$ choosing new elements of Θ to form Θ'_n extending Θ_n and defining them on $\mathcal{L}_n$ so that $\Theta'_n \restriction \mathcal{L}_n$ is an usl table for $\mathcal{L}_n$ containing Θ_n and the required meet interpolants for every such x, y, z, α and β. We then apply Proposition hominterp 10.3.6 successively for each $\hat{\mathcal{L}} \subseteq_{lsl} \mathcal{L}_n$ and each $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \Theta_n$ such that $\forall w \in \hat{\mathcal{L}}(\alpha_0 \equiv_w \alpha_1 \rightarrow \beta_0 \equiv_w \beta_1)$ to get larger subset Θ''_n of Θ which we also define on $\mathcal{L}_n$ so as to have an usl table $\Theta''_n \restriction \mathcal{L}_n$ for $\mathcal{L}_n$ that has the required homogeneity interpolants and $\hat{\mathcal{L}}$-homomorphisms from Θ_n into Θ''_n for every such $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \Theta_n$. Finally, we apply Proposition extend 10.3.4 to define the elements of Θ''_n on $\mathcal{L}_{n+1}$ and further enlarge it to our desired finite $\Theta_{n+1} \subseteq \Theta$ with all its new elements also defined on $\mathcal{L}_{n+1}$ so as to have an usl table of $\mathcal{L}_{n+1}$ with all the properties required by the Theorem. It is now immediate from the definitions that the union Θ of the Θ_n is an usl table of $\mathcal{L}$. ■

Notation 10.3.2 *If a finite $\hat{\mathcal{L}}$ is a slsl of $\mathcal{L}$, $\hat{\mathcal{L}} \subseteq_{lsl} \mathcal{L}$, and $x \in \mathcal{L}$ then we let $\hat{x}$ denote the least element of $\hat{\mathcal{L}}$ above x. The desired element of $\hat{\mathcal{L}}$ exists because $\hat{\mathcal{L}}$ is a slsl of $\mathcal{L}$ and so the meet (in $\hat{\mathcal{L}}$ or, equivalently, in $\mathcal{L}$) of $\{u \in \hat{\mathcal{L}} | x \leq u\}$ is in $\hat{\mathcal{L}}$ and is the desired $\hat{x}$. As $\hat{\mathcal{L}}$ is finite it is also a lattice but join in $\hat{\mathcal{L}}$ may not agree with that in $\mathcal{L}$. We denote them by $\vee_{\hat{\mathcal{L}}}$ and $\vee_{\mathcal{L}}$ respectively when it is necessary to make this distinction.*

basichat **Lemma 10.3.3** *With the notation as above, $\hat{x} = x$ for $x \in \hat{\mathcal{L}}$ and so it is an idempotent operation. If $x \leq y$ are in $\mathcal{L}$ then $\hat{x} \leq \hat{y}$. If $x \vee_{\mathcal{L}} y = z$ are in $\mathcal{L}$ then $\hat{z} = \hat{x} \vee_{\hat{\mathcal{L}}} \hat{y}$.*

Proof. The first two assertions follow immediately from the definition of $\hat{x}$. The third is only slightly less immediate: $x, y \leq x \vee_{\mathcal{L}} y = z$ and so by the second assertion, $\hat{x}, \hat{y} \leq \hat{z}$ and so $\hat{x} \vee_{\hat{\mathcal{L}}} \hat{y} \leq \hat{z}$. For the other direction, note that as $x \leq \hat{x}$, $y \leq \hat{y}$, we have that $z = x \vee_{\mathcal{L}} y \leq \hat{x} \vee_{\mathcal{L}} \hat{y} \leq \hat{x} \vee_{\hat{\mathcal{L}}} \hat{y} \in \hat{\mathcal{L}}$ and so $\hat{z} \leq \hat{x} \vee_{\hat{\mathcal{L}}} \hat{y}$. ■

extend **Proposition 10.3.4** *If Θ is a finite usl table for $\hat{\mathcal{L}} \subseteq_{lsl} \mathcal{L}$ (finite) then there are extensions for each $\alpha \in \Theta$ to maps with domain $\mathcal{L}$ and finitely many further functions β with domain $\mathcal{L}$ such that adding them on to our extensions of the $\alpha \in \Theta$ provides an usl table Θ' of $\mathcal{L}$ with $\Theta \subseteq \Theta' \restriction \hat{\mathcal{L}}$. Moreover, these extensions can be found uniformly recursively in the given data (Θ, $\hat{\mathcal{L}}$ and $\mathcal{L}$).*

Proof. For $\alpha \in \Theta$ and $x \in \mathcal{L}$ set $\alpha(x) = \alpha(\hat{x})$. We first check that we have maintained the order and join properties required of an usl representation. If $x \leq y$ are in $\mathcal{L}$, $\alpha, \beta \in \Theta$

and $\alpha \equiv_y \beta$ then by definition $\alpha \equiv_{\hat{y}} \beta$ and so $\alpha \equiv_{\hat{x}} \beta$ as $\hat{x} \leq \hat{y}$ by Lemma basichat 10.3.3 and Θ's being an usl table of $\hat{\mathcal{L}}$. Thus, by definition, $\alpha \equiv_x \beta$ as required.

Next, if $x \vee_{\mathcal{L}} y = z$ are in $\mathcal{L}$ and $\alpha \equiv_{x,y} \beta$ we wish to show that $\alpha \equiv_z \beta$. Again by definition $\alpha \equiv_{\hat{x},\hat{y}} \beta$. By Lemma basichat 10.3.3, $\hat{x} \vee_{\hat{\mathcal{L}}} \hat{y} = \hat{z}$, so by Θ being an usl table for $\hat{\mathcal{L}}$, $\alpha \equiv_{\hat{z}} \beta$ and so by definition, $\alpha \equiv_z \beta$.

All that remains is to show that we can add on new maps with domain $\mathcal{L}$ that provide witnesses for the differentiation property for elements of $\mathcal{L} - \hat{\mathcal{L}}$ while preserving the order and join properties. This is a standard construction. For each pair $x \nleq y$ (in $\mathcal{L}$ but not both in $\hat{\mathcal{L}}$) in turn we add on new elements $\alpha_{x,y}$ and $\beta_{x,y}$ with all new and distinct values at each $z \in \mathcal{L}$ except that they agree on all $z \leq x$ (and at 0, of course, have value 0). These new elements obviously provide the witnesses required for the differentiation property for an usl representation. It is easy to see that they also cause no damage to the order or join properties. There are no new nontrivial instances of congruences between them and the old ones in Θ (extended to $\mathcal{L}$). Among the new elements the only instances to consider are ones between $\alpha_{x,y}$ and $\beta_{x,y}$ for the same pair x, y and for lattice elements z less than or equal to x. As $\alpha_{x,y} \equiv_z \beta_{x,y}$ for all $z \leq x$, the order and join properties are immediate. ■

meetinterp **Proposition 10.3.5** *If $\alpha, \beta \in \Theta$, an usl table for a finite lattice $\mathcal{L}$, $\alpha \equiv_z \beta$ and $x \wedge y = z$ in $\mathcal{L}$ then there are $\gamma_0, \gamma_1, \gamma_2$ such that $\alpha \equiv_x \gamma_0 \equiv_y \gamma_1 \equiv_x \gamma_2 \equiv_y \beta$ and $\Theta \cup \{\gamma_0, \gamma_1, \gamma_2\}$ is still an usl table for $\mathcal{L}$. Moreover, these extensions can be found uniformly recursively in the given data.*

Proof. If $x \leq y$, there is nothing to be proved. Otherwise, the interpolants can be defined by letting $\gamma_0(w)$ be $\alpha(w)$ for $w \leq x$ and new values for $w \nleq x$; $\gamma_1(w) = \gamma_0(w)$ for $w \leq y$ and new values otherwise; and $\gamma_2(w) = \beta(w)$ for $w \leq y$, $\gamma_2(w) = \gamma_1(w)$ if $w \leq x$ but $w \nleq y$ and new otherwise. ■

hominterp **Proposition 10.3.6** *If $\hat{\mathcal{L}} \subseteq_{lsl} \mathcal{L}$, a finite lattice, and Θ is an usl table for $\mathcal{L}$ with $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \Theta$ such that $\forall w \in \hat{\mathcal{L}}(\alpha_0 \equiv_w \alpha_1 \rightarrow \beta_0 \equiv_w \beta_1)$, then there is an usl table $\tilde{\Theta} \supseteq \Theta$ for $\mathcal{L}$ with $\gamma_0, \gamma_1 \in \tilde{\Theta}$ and $\hat{\mathcal{L}}$ homomorphisms $f, g, h : \Theta \rightarrow \tilde{\Theta}$ such that $f : \alpha_0, \alpha_1 \mapsto \beta_0, \gamma_1$, $g : \alpha_0, \alpha_1 \mapsto \gamma_0, \gamma_1$ and $h : \alpha_0, \alpha_1 \mapsto \gamma_0, \beta_1$. Moreover, these extensions can be found uniformly recursively in the given data.*

Proof. For each $\alpha \in \Theta$ and $x \in \mathcal{L}$ we set $f(\alpha)(x) = \beta_0(x)$ if $\alpha \equiv_{\hat{x}} \alpha_0$ and otherwise we let it be a new number that depends only on $\alpha(\hat{x})$, e.g. $\alpha(\hat{x})^*$. Note that which case of the definition applies for $f(\alpha)(x)$ depends only on $\alpha(\hat{x})$ and it can be an "old" value (i.e. one of some $\beta \in \Theta$) only in the first case. Thus, for $\alpha, \beta \in \Theta$,

$$\text{(a) } \alpha \equiv_{\hat{x}} \beta \Leftrightarrow f(\alpha) \equiv_x f(\beta) \text{ and (b) } f(\alpha) \equiv_x \beta \Rightarrow \alpha \equiv_{\hat{x}} \alpha_0 \Rightarrow f(\alpha) \equiv_x \beta_0. \qquad (10.1)$$ 1

Let $\Theta_1 = \Theta \cup f[\Theta]$. We claim that Θ_1 is an usl table for $\mathcal{L}$ and f is an $\hat{\mathcal{L}}$-homomorphism from Θ into Θ_1. That f is an $\hat{\mathcal{L}}$-homomorphism is immediate from the first clause in

(10.1) and the fact (Lemma 10.3.3) that $\hat{x} = x$ for $x \in \hat{\mathcal{L}}$. We next check that Θ_1 satisfies the properties required of an usl representation. Of course, $f(\alpha)(0) = 0$ by definition for every α and differentiation is automatic as it extends Θ.

First, to check the order property for Θ_1 we consider any $x \leq y$ in $\mathcal{L}$. As Θ is already an usl table for $\mathcal{L}$, it suffices to consider two cases for the pair of elements of Θ_1 which are given as congruent modulo y and show that in these two cases they are also congruent modulo x. The two cases are that (a) both are in $f[\Theta]$ and that (b) one is in $f[\Theta]$ and the other in Θ. Thus it suffices to consider any $\alpha, \beta \in \Theta$, assume that (a) $f(\alpha) \equiv_y f(\beta)$ or (b) $f(\alpha) \equiv_y \beta$ and prove that (a) $f(\alpha) \equiv_x f(\beta)$ and (b) $f(\alpha) \equiv_x \beta$, respectively. For (a), we have by (10.1) that $\alpha \equiv_{\hat{y}} \beta$ and so by the order property for Θ, $\alpha \equiv_{\hat{x}} \beta$. Thus $f(\alpha) \equiv_x f(\beta)$ by definition as required. As for (b), (10.1) tells us here that $\alpha \equiv_{\hat{y}} \alpha_0$ and $\beta \equiv_y f(\alpha) \equiv_y \beta_0$ (and therefore $\beta \equiv_x \beta_0$). Now by Lemma 10.3.3 $\alpha \equiv_{\hat{x}} \alpha_0$ so $f(\alpha) \equiv_x \beta_0$ and so $f(\alpha) \equiv_x \beta$ as required.

Next we verify the join property for $x \vee y = z$ in $\mathcal{L}$ and two elements of Θ_1 (not both in Θ) in the same two cases. For (a) we have that $f(\alpha) \equiv_{x,y} f(\beta)$ and so as above $\alpha \equiv_{\hat{x},\hat{y}} \beta$. Now by the join property in Θ and Lemma 10.3.3, $\alpha \equiv_{\hat{z}} \beta$ and so $f(\alpha) \equiv_z f(\beta)$ as required. For (b) using (10.1b) and Lemma 10.3.3 again we have that $f(\alpha) \equiv_{x,y} \beta \Rightarrow \alpha \equiv_{\hat{x},\hat{y}} \alpha_0 \Rightarrow \alpha \equiv_{\hat{z}} \alpha_0 \Rightarrow f(\alpha) \equiv_z \beta_0$ while it also tells us that $\beta \equiv_{x,y} f(\alpha) \equiv_{x,y} \beta_0$ as required. Note that clearly $f(\alpha_0) = \beta_0$. We let $\gamma_1 = f(\alpha_1)$ and so have the first function and (partial) extension of Θ required in the Proposition.

We now define h on Θ_1 as we did f on Θ using α_1 and β_1 in place of α_0 and β_0, respectively: $h(\alpha)(x) = \beta_1(x)$ if $\alpha \equiv_{\hat{x}} \alpha_1$ and otherwise we let it be a new number that depends only on $\alpha(\hat{x})$, e.g. $\alpha(\hat{x})^{**}$. Let $\Theta_2 = \Theta_1 \cup h[\Theta_1]$. As above, Θ_2 is an usl table for $\mathcal{L}$ and h is an $\hat{\mathcal{L}}$-homomorphism from Θ_1 (and so Θ) into Θ_2 taking α_1 to β_1. We let $\gamma_0 = h(\alpha_0)$ and so have the third function and (partial) extension of Θ required in the Proposition. As above in (10.1), we have for any $\alpha, \beta \in \Theta_1$ and $x \in \mathcal{L}$,

$$\text{(a) } \alpha \equiv_{\hat{x}} \beta \Leftrightarrow h(\alpha) \equiv_x h(\beta) \text{ and (b) } h(\alpha) \equiv_x \beta \Rightarrow \alpha \equiv_{\hat{x}} \alpha_1 \Rightarrow h(\alpha) \equiv_x \beta_1. \qquad (10.2) \quad \boxed{2}$$

Applying the second clause to $\gamma_0 = h(\alpha_0)$ and first to any $\beta \in \Theta_1$ and then, in particular to γ_1 we have

$$\text{(a) } \gamma_0 \equiv_x \beta \Rightarrow \alpha_0 \equiv_{\hat{x}} \alpha_1 \Rightarrow f(\alpha_1) = \gamma_1 \equiv_x \beta_0 \text{ and (b) } \gamma_0 \equiv_x \gamma_1 \Leftrightarrow \alpha_0 \equiv_{\hat{x}} \alpha_1. \qquad (10.3) \quad \boxed{3}$$

To see the right to left direction of the second clause, note that $\alpha_0 \equiv_{\hat{x}} \alpha_1$ implies that $\gamma_0 \equiv_x \beta_1$ and $\gamma_1 \equiv_x \beta_0$ by the definitions of h and f, respectively, while it also implies that $\beta_0 \equiv_{\hat{x}} \beta_1$ by the basic assumption of the Proposition. Thus, as Θ is an usl table of $\mathcal{L}$ and $x \leq \hat{x}$, $\beta_0 \equiv_x \beta_1$ and $\gamma_0 \equiv_x \gamma_1$.

Finally, we define g on $\alpha \in \Theta_2$ by setting $g(\alpha)(x) = \gamma_0(x)$ if $\alpha \equiv_{\hat{x}} \alpha_0$. If $\alpha \not\equiv_{\hat{x}} \alpha_0$ but $\alpha \equiv_{\hat{x}} \alpha_1$ then $g(\alpha)(x) = \gamma_1(x)$. Otherwise, we let $g(\alpha)(x)$ be a new number that depends only on $\alpha(\hat{x})$, e.g. $\alpha(\hat{x})^{***}$. Note that if $\alpha \equiv_{\hat{x}} \alpha_1$ then we always have $g(\alpha) \equiv_x \gamma_1$ as if $\alpha \equiv_{\hat{x}} \alpha_0$ as well then, by (10.3b), $\gamma_0 \equiv_x \gamma_1$. Thus $g(\alpha_0) = \gamma_0$ and $g(\alpha_1) = \gamma_1$ as required. It is also obvious that g is an $\hat{\mathcal{L}}$-homomorphism of Θ_2 (and so Θ) into $\Theta_3 = \Theta_2 \cup g[\Theta_2]$

as by definition and Lemma 10.3.3, $\alpha \equiv_{\hat{x}} \beta \Rightarrow g(\alpha) \equiv_{\hat{x}} g(\beta)$ for any $x \in \mathcal{L}$. Indeed, for any $\alpha, \beta \in \Theta_2$ and $x \in \mathcal{L}$

$$\alpha \equiv_{\hat{x}} \beta \Leftrightarrow g(\alpha) \equiv_x g(\beta). \tag{10.4}$$ 4

To see the right to left direction here, note that if either of $g(\alpha)$ or $g(\beta)$ is new for g at x (i.e. of the form $\delta(\hat{y})^{***}$) then clearly both are. In this case, $\alpha \equiv_{\hat{x}} \beta$ by definition. Otherwise, either they are both congruent to α_0 or both to α_1 and so congruent to each other mod $\hat{x}$. The point here is that if one is congruent to α_0 and the other to α_1 but not α_0 at $\hat{x}$ then by definition $\gamma_0 \equiv_x \gamma_1$ and so by (10.3b), $\alpha_0 \equiv_{\hat{x}} \alpha_1$ for a contradiction.

Thus we only need to verify that Θ_3 is an usl table of $\mathcal{L}$. We consider any $\alpha, \beta \in \Theta_2$ and divide the verifications into cases (a) and (b) as before with the former considering $g(\alpha)$ and $g(\beta)$ and the latter $g(\alpha)$ and β. These cases may then be further subdivided.

We begin with the order property and so $x \leq y$ in $\mathcal{L}$.

(a) If $g(\alpha) \equiv_y g(\beta)$ then, by (10.4), $\alpha \equiv_{\hat{y}} \beta$ and so $\alpha \equiv_{\hat{x}} \beta$ as $\hat{x} \leq \hat{y}$ (Lemma 10.3.3) and Θ_2 is an usl table of $\mathcal{L}$. Thus, again by (10.4) $g(\alpha) \equiv_x g(\beta)$ as required.

(b) If $g(\alpha) \equiv_y \beta$ then by definition they are congruent modulo y to γ_i (for some $i \in \{0, 1\}$) and α is congruent to α_i at $\hat{y}$. Thus $\alpha \equiv_{\hat{x}} \alpha_i$ as $\hat{x} \leq \hat{y}$ and Θ_2 is an usl table so $g(\alpha) \equiv_x \gamma_i$ by definition. Similarly, as $x \leq y$, $\beta \equiv_x \gamma_i$ as well.

Now for the join property for $x \vee y = z$ in $\mathcal{L}$.

(a) If $g(\alpha) \equiv_{x,y} g(\beta)$ then, as above, $\alpha \equiv_{\hat{x},\hat{y}} \beta$. As $\hat{x} \vee \hat{y} = \hat{z}$ by Lemma 10.3.3 and Θ_2 is an usl representation, $\alpha \equiv_{\hat{z}} \beta$ and so by (10.4) $g(\alpha) \equiv_z g(\beta)$ as required.

(b) If $g(\alpha) \equiv_{x,y} \beta$ then again $\alpha \equiv_{\hat{x}} \alpha_i$ and $\alpha \equiv_{\hat{y}} \alpha_j$ for some $i, j \in \{0, 1\}$ and $g(\alpha) \equiv_x \beta \equiv_x \gamma_i$ while $g(\alpha) \equiv_y \beta \equiv_y \gamma_j$. If $i = j$ then $\alpha \equiv_{\hat{x},\hat{y}} \alpha_i$ and so $\alpha \equiv_{\hat{z}} \alpha_i$ and $g(\alpha) \equiv_z \gamma_i \equiv_z \beta$ as required.

On the other hand, suppose, without loss of generality, that $(*)$ $\alpha \equiv_{\hat{x}} \alpha_0$ and so $\beta \equiv_x g(\alpha) \equiv_{\hat{x},x} \gamma_0 = h(\alpha_0)$ while $\alpha_0 \not\equiv_{\hat{y}} \alpha \equiv_{\hat{y}} \alpha_1$ and so $\beta \equiv_y g(\alpha) \equiv_{\hat{y},y} \gamma_1 = f(\alpha_1)$. If $\beta \in \Theta_1$ then by (10.4a) $\alpha_0 \equiv_{\hat{x}} \alpha_1$ and so $\alpha \equiv_{\hat{x}} \alpha_1$. As our assumption is that $\alpha \equiv_{\hat{y}} \alpha_1$ we have (by the join property in Θ_2) that $\alpha \equiv_{\hat{z}} \alpha_1$ and so $g(\alpha) \equiv_z \gamma_1$. As $\alpha_0 \equiv_{\hat{x}} \alpha_1$ (10.3b) tells us that $\gamma_0 \equiv_x \gamma_1$. Our assumptions then say that $\beta \equiv_{x,y} \gamma_1$ and so $\beta \equiv_z \gamma_1$ as required. Thus we may assume that $\beta = h(\delta)$ for some $\delta \in \Theta_1$.

We now have $h(\delta) = \beta \equiv_x g(\alpha) \equiv_x \gamma_0 = h(\alpha_0) \in \Theta_1$ and so by (10.2a) applied to $h(\delta) \equiv_x h(\alpha_0)$ with δ for α and α_0 for β we see that $\delta \equiv_{\hat{x}} \alpha_0$. We also have $h(\delta) = \beta \equiv_y g(\alpha) \equiv_{\hat{y},y} \gamma_1 = f(\alpha_1)$. Applying (10.2b) to $h(\delta) \equiv_y \gamma_1$ with δ for α and $\gamma_1 \in \Theta_1$ for β, we see that $\delta \equiv_{\hat{y}} \alpha_1$and $h(\delta) \equiv_y \beta_1$ and so $\beta_1 \equiv_y \gamma_1 = f(\alpha_1)$. Now applying (10.1b) with α_1 for α and $\beta_1 \in \Theta$ for β, we have that $\alpha_1 \equiv_{\hat{y}} \alpha_0$. As this contradicts $(*)$, we are done. ■

2qtth 10.4 Decidability of two quantifier theory

3qtth 10.5 Undecidability of three quantifier theory.

Also two quantifier with $\vee$ and $\wedge$.?

Other results establishing borderlines in other languages e.g. with jump? If so in different chapter/section?

comments on what known below $\mathbf{0}'$

Chapter 11

Π^0_1 Classes

11.1 Binary trees

We now return to the basic our basic notion of a tree as a downward closed subset of $\mathbb{N}^{<\omega}$. In this context we use T_σ to denote the subtree of T consisting of all strings ρ compatible with σ: $T_\sigma = \{\rho|\rho \subseteq \sigma$ or $\sigma \subseteq \rho\}$. Recall that the sets of paths in such trees are the closed sets in Baire space $\mathbb{N}^\mathbb{N}$. In this chapter we are primarily concerned with infinite binary trees, i.e. the infinite downward closed subsets T of $2^{<\omega}$. We endow each binary tree with a left to right partial order as well as the order of extension. It is specified by the lexicographic order on strings so σ is to the left of τ, $\sigma <_L \tau$ if $\sigma(n) < \tau(n)$ for the least n such that $\sigma(n) \neq \tau(n)$ if there is one. (This order extends in the obvious way to one of $2^\mathbb{N}$ which we also call the left to right or lexicographic order.) The sets of paths $[T] = \{A \in 2^\mathbb{N} : \forall n(A \upharpoonright n \in T)\}$ through these trees are precisely the nonempty closed subsets of Cantor space, $2^\mathbb{N}$.

Exercise 11.1.1 *For any binary tree T, $[T]$ is a closed set in Cantor space.*

??Prove??

To see that every closed subset of $2^\mathbb{N}$ is of the form $[T]$ for some tree T, consider the open sets in $2^\mathbb{N}$. They are all unions of basic (cl)open sets of the form $[\sigma] = \{f \in 2^\mathbb{N}|\sigma \subseteq f\}$ for $\sigma \in 2^{<\omega}$. So given any closed set $\mathcal{C}$ its complement $\bar{\mathcal{C}}$ is a union of such neighborhoods. Let $T = \{\sigma|[\sigma] \subsetneqq \bar{\mathcal{C}}\} = \{\sigma|[\sigma] \cap \mathcal{C} \neq \emptyset\}$. It is clear that T is downward closed. If $f \in \mathcal{C}$ and $\sigma \subseteq f$ then clearly $\sigma \in T$ and so $f \in [T]$. On the other hand if $f \in [T]$ and $\sigma \subseteq f$ then $\sigma \in T$ and so the closed set $[\sigma] \cap \mathcal{C} \neq \emptyset$. As Cantor space is compact $\cap\{[\sigma] \cap \mathcal{C}|\sigma \subseteq f\}$ is nonempty and only f can be in it so $f \in \mathcal{C}$ as required. Note that, by König's lemma (Lemma $\overset{\text{KL}}{4.2.4}$), $\mathcal{C}$ is nonempty if and only if T is infinite.

??Move this material to Trees section and recall here??

In this chapter we want to investigate the recursive versions of these two notions.

Definition 11.1.2 *A class $\mathcal{C} \subseteq 2^\mathbb{N}$ is* effectively closed *if it is of the form $[T]$ for a recursive binary tree T.*

We can also characterize the effectively closed sets in terms of the complexity of their definition. We use the same notation based on the arithmetic hierarchy for classes of sets or functions as we did for individual sets and functions.??say more now or before Go back and check definitions for Σ^A_n especially Σ_0 and how interpret for σ in place of A ...bounded quantifiers??

Definition 11.1.3 *A class $\mathcal{C} \subseteq 2^{\mathbb{N}}$ of sets is Σ_n (Π_n) if there is a Σ_n (Π_n) formula $\varphi(X)$ with one free set variable X such that $\mathcal{C} = \{A | \mathbb{N} \vDash \varphi(A)\}$. Similarly for classes $\mathcal{F} \subseteq \mathbb{N}^{\mathbb{N}}$ of functions and formulas with one free function variable.*

The primary connection with trees is the following Proposition.

pi01trees **Proposition 11.1.4** *The Π^0_1 classes of sets are precisely the sets of paths through recursive binary trees. Again, the nonempty classes correspond to the infinite recursive binary trees. Moreover, there is a recursive procedure that takes an index for a Π^0_1 formula to one for a recursive tree T such that $[T]$ is the corresponding Π^0_1 class.*

Proof. If T is a recursive binary tree then $[T] = \{A \in 2^{\mathbb{N}} : \forall n(A \upharpoonright n \in T)\}$ is clearly a Π^0_1 class. If T is infinite, $[T]$ is nonempty by König's lemma while if T is finite $[T]$ is clearly empty. For the other direction consider any Π^0_1 class $\mathcal{P} = \{A : \forall x R(A, x)\}$ for a Σ^A_0 relation R. Let $T = \{\sigma \in 2^{<\omega} | \neg(\exists x < |\sigma|) \neg R(\sigma, x)\}$ where we understand that we are thinking of σ as representing an initial segment of A. Formally we replace $t \in A$ by $\sigma(t) = 1$ and declare the formula $R(\sigma, x)$ false if some term $t > |\sigma|$ occurs in it as described in ??. It is then immediate that $P = [T]$ and that an index for T as recursive function is given uniformly in the index for R as a Σ^A_0 formula. If P is nonempty, T has an infinite path and so is itself infinite. Otherwise, T is finite. ■

Exercise 11.1.5 *The Π^0_1 classes of functions are precisely the sets of paths through recursive trees (on $\mathbb{N}^{<\omega}$).*

We can now index the Π^0_1 classes (of sets) by either the indices of the Π^0_1 formulas or of the trees derived from them as in the proof of Proposition pi01trees 11.1.4 as partial recursive functions which are actually total. A natural question then is how hard is to tell if a recursive tree is infinite or a Π^0_1 class is nonempty. It might seem at first that these properties are Π^0_2 and so only recursive in $0''$. If we know that the tree is recursive as we do for the trees derived uniformly from Π^0_1 classes, however, then the question is actually uniformly (on indices) recursive in $0'$. This observation depends on the compactness of Cantor space and plays a crucial role in almost every argument in the rest of this chapter.

fin0' **Lemma 11.1.6** *If T is a recursive binary tree (say with index i so $T = \Phi_i$) then T being finite is a Σ_1 property (of i). Thus we can decide if T is finite or infinite recursively in $0'$. Indeed, T is finite if and only if there is an n such that $\sigma \notin T$ for every σ of length n.*

Proof. Clearly, T is finite if and only if there is an n such that $\sigma \notin T$ for every σ of length n. Clearly this is a Σ_1 property for any recursive binary tree and the associated Σ_1 formula is given uniformly in a recursive index for T. ■

While deciding if a given recursive binary tree is infinite or a Π^0_1 class nonempty requires $0'$, we can actually make a recursive list of the nonempty Π^0_1 classes and so one of corresponding infinite recursive binary trees (up to the set of paths on T).

recindpi01 **Exercise 11.1.7** *There is a uniformly recursive list of the nonempty Π^0_1 classes in the sense that there is a recursive set Q such that, for each $e \in Q$, Φ_e is (the characteristic function of) an infinite binary tree T_e and for every nonempty Π^0_1 class $\mathcal{C}$ there is an e such that $\mathcal{C} = [\Phi_e] = [T_e]$. Hint: For each e consider the r.e. set W_e viewing its elements as binary strings σ. We now form a recursive tree T_e by putting in the empty string at stage 0 and then at stage $s > 0$ exactly those strings τ of length s with no $\sigma \subseteq \tau$ in $W_{e,s}$ unless there are none (equivalently $\cup\{[\sigma]|\sigma \in W_{e,s}\} = 2^{\mathbb{N}}$), in which case we declare all immediate successors of strings in T_e of length $s-1$ to be in T_e as well. Note that T_e is uniformly recursive. For one direction prove that each T_e is infinite (and so $[T_e]$ is a nonempty Π^0_1 class). For the other direction, if $\mathcal{C}$ is a nonempty Π^0_1 class then the set $\{\sigma|[\sigma] \cap \mathcal{C} = \emptyset\}$ is r.e. and so equal to some W_e. Now show that $[T_e] = \mathcal{C}$.*

We now present some important Π^0_1 classes.

Example 11.1.8 *$DNR_2 = \{f \in 2^{\mathbb{N}} : f$ is $DNR\}$. Recall that DNR means $f(e) \neq \Phi_e(e)$. In other words, $\forall e \forall s \neg(f(e) = \Phi_{e,s}(e))$. Thus, DNR_2 is a Π^0_1 class.*

Example 11.1.9 *Let H be any recursively axiomatizable consistent theory. The class $\mathcal{C}_H$=$\{f \in 2^{\mathbb{N}} : f$ is a complete extension of $H\}$ is a Π^0_1 class. By the assertion that f "is a complete extension of H" we mean that we have a recursive coding (Gödel numbering) φ_n of the sentences of H such that $T_f = \{\varphi_n|f(n) = 1\}$ is deductively closed, contains all the axioms of H and is consistent in the sense that there is no φ such that f assigns* 1 *(true) to both φ and $\neg\varphi$. The only point to make about this being a Π^0_1 class is perhaps the requirement that T_f be deductively closed. This says that for all finite sets Φ of sentences and each sentence φ_k and proof p, if p is a proof that $\Phi \vdash \varphi$ and $f(n) = 1$ for every $\varphi_n \in \Phi$ then $f(k) = 1$.*

Example 11.1.10 *If A, B are disjoint r.e. sets, then the class $S(A, B) = \{C : C \supset A \;\&\; C \cap B = \emptyset\}$ of* separating sets *C (for the pair (A, B)) is a Π^0_1 class as is obvious from its definition: $S = \{C : \forall n(n \in A \to n \in C \;\&\; n \in B \to n \notin C)\}$. Since $A, B \in \Sigma_1$ this is a Π^0_1 formula.*

We can view a Π^0_1 class as the solution set to the problem of finding an f that satisfies the defining condition for the class. Equivalently, the problem is finding a path f through the corresponding tree T. For the above examples the problems are to construct a DNR_2 function, a complete consistent extension of H and a separating set for A

and B, respectively. If we choose our theory H and our disjoint r.e. sets A and B correctly then the three problems and so the Π^0_1 classes (and the $[T]$ for the corresponding trees) are equivalent in the sense that a solution to (path through) any one of them computes a solution for (path in) each of the others. Suitable choices for H and (A, B) are Peano arithmetic, PA, ??define before?? and (V_0, V_1) where $V_0 = \{e : \Phi_e(e) = 0\}$ and $V_1 = \{e : \Phi_e(e) = 1\}$. For these choices, the problems are also universal in the sense that a solution to any one of them computes a path through any infinite recursive binary tree and hence a solution to any problem specified by a nonempty Π^0_1 class.

Theorem 11.1.11 *If T is an infinite recursive binary tree and f is a member of any of the three Π^0_1 classes DNR_2, $\mathcal{C}_{PA}$ or $S(V_0, V_1)$ described above then there is a path $g \in [T]$ with $g \leq_T f$.*

Proof. We first prove the theorem for $S(V_0, V_1)$. Suppose T is an infinite recursive binary tree. We begin by defining disjoint r.e. sets A and B such that any $f \in S(A, B)$ computes a path in T. We then show how to compute a path in (any) $S(A, B)$ from one in $S(V_0, V_1)$.

We know that $\{\sigma | T_\sigma \text{ is finite}\}$ is r.e. so suppose it is W_e. We let $A_0 = \{\sigma | \exists s(\sigma\hat{\ }0 \in W_{e,s}$ & $\sigma\hat{\ }1 \in W_{e,s}\}$ (the σ such that we "see" that $T_{\sigma\hat{\ }0}$ is finite before we "see" that $T_{\sigma\hat{\ }1}$ is finite) and $A_1 = \{\sigma | \exists s(\sigma\hat{\ }1 \in W_{e,s}$ & $\sigma\hat{\ }0 \in W_{e,s}\}$ (the σ such that we "see" that $T_{\sigma\hat{\ }1}$ is finite before we "see" that $T_{\sigma\hat{\ }0}$ is finite). It is clear that $A_0 \cap A_1 = \emptyset$. Let $C \in S(A_0, A_1)$ and define D a path in T by recursion. We begin with $\emptyset \in D$. If $\sigma \in D$ then we put $\sigma\hat{\ }C(\sigma)$ into D. We now argue by induction that if $\sigma \in D$ then T_σ is infinite: If T_σ is infinite then at least one of $T_{\sigma\hat{\ }0}$ and $T_{\sigma\hat{\ }1}$ is infinite. If both are infinite there is nothing to prove so suppose that $T_{\sigma\hat{\ }0}$ is finite but $T_{\sigma\hat{\ }1}$ is infinite. In this case, it is clear from the definition that $\sigma \in A_0$ and so $C(\sigma) = 1$ and we put $\sigma\hat{\ }1$ into D to verify the induction hypothesis. In the other case, $\sigma \in A_1$, $C(\sigma) = 0$ and we put $\sigma\hat{\ }0$ into D with $T_{\sigma\hat{\ }0}$ infinite as required.

Now we see how to compute a $C \in S(A_0, A_1)$ from any $D \in S(V_0, V_1)$. By the $s{-}m{-}n$ theorem ?? there is a recursive functions h such that $\forall n(n \in A_i \Leftrightarrow h(n) \in V_i)$. We now let $C(n) = D(h(n)$. It is easy to see that $C \in S(A_0, A_1)$ as required. Thus $S(V_0, V_1)$ is universal in the desired sense.

We now only have to prove that we can compute a member of $S(V_0, V_1)$ from any DNR_2 function f and from any complete extension P of PA. For the first, simply note that if $f \in DNR_2$ then $f \in S(V_0, V_1)$: If $e \in V_0$ then $\Phi_e(e) = 0$ and so $f(e) = 1$ as required. On the other hand, $e \in V_1$ then $\Phi_e(e) = 1$ and so $f(e) = 0$ as required.

Finally, suppose P is complete extension of PA. Define $C(n) = 1$ if P declares the sentence $\exists s(n \in V_{0,s}$ & $\forall t < s(n \notin V_{1,s}))$ to be true and 0 otherwise. Note that if $n \in V_0$ then there is a least $s \in \mathbb{N}$ such that $n \in V_{0,s}$. This fact is then provable in PA (computation is essentially a proof). Similarly, for each $t < s$, $n \notin V_{1,t}$ since $n \in V_0$ and so $C(n) = 1$ as required. On the other hand, if $n \in V_1$ then there is a least $s \in \mathbb{N}$ such that $n \in V_{1,s}$ and for each $t < s$, $n \notin V_{0,t}$ since $n \in V_1$ and so PA proves that

$\exists s(n \in V_{1,s}\ \&\ \forall t < s(n \notin V_{0,s}))$. As P is a consistent extension of PA, it cannot then prove that $\exists s(n \in V_{0,s}\ \&\ \forall t < s(n \notin V_{1,s}))$ and so $C(n) = 0$ as required.

Exercise 11.1.12 *Show that the degree classes* $\mathbf{DNR}_2$*,* $\mathbf{C}_{PA}$ *and* $\mathbf{S}(\mathbf{V}_0, \mathbf{V}_1)$ *consisting of the degrees in each of the corresponding* Π^0_1 *classes are all the same.*

■

As every DNR function is obviously nonrecursive (Proposition dnrnotrec 3.0.19), none of these three classes have recursive members. So in particular there are no recursive complete extension of PA and there is no recursive separating set for (V_0, V_1).

Thinking of Π^0_1 classes as problems that ask for solutions, the natural question is how complicated must solutions be or how simple can they be. In the (in some sense uninteresting) case that there is only one path in T (or only finitely many) we can say everything about their degrees.

Proposition 11.1.13 *If a recursive binary tree T has single path that path is recursive. In fact, any isolated path ??define?? on a recursive tree is recursive.*

In general for arbitrary T one easy answer to the question is that there are always solutions recursive in $0'$.

rec0' **Exercise 11.1.14** *Show that every nonempty* Π^0_1 *class has a member recursive in* $0'$*. Hint: it is immediate for the separating classes.*

It is not hard to say a bit more.

redegree **Proposition 11.1.15** *Every infinite recursive binary tree T has a path of r.e. degree. In fact, the leftmost path P in T has r.e. degree.*

We, in fact, can significantly improve the result of Exercise rec0' 11.1.14. The Low Basis Theorem below (Theorem lowb 11.1.18) gives the best answer with the notion of simplicity of the desired solution measured by its jump class. It is called a *basis theorem* as we say that a class $\mathcal{C}$ is a *basis* for a collection of problems, (sets) if every problem (set) in the collection has a solution (member) in $\mathcal{C}$. Theorem 0'domb 11.1.19 gives another basis result in terms of domination properties and Theorem pi01coneav 11.1.21 one in terms of solutions not computing given (nonrecursive) sets.

To prove each of these theorems we use the notion of forcing $\mathcal{P}$ whose conditions are basically infinite recursive binary trees T with usual notion of subtree as extension (simply a subset). To make the definition of our required function V recursive, we explicitly specify a stem τ for each tree such that every $\rho \in T$ is compatible with σ. Thus our conditions p are pairs (T, τ) with T an infinite recursive binary tree and $\tau \in T$ such that $(\forall \rho \in T)(\rho \subseteq \tau$ or $\tau \subseteq \rho)$. We say that $(T, \tau) \leq_{\mathcal{P}} (S, \sigma)$ if $T \subseteq S$ and $\tau \supseteq \sigma$. Of course, $V((T, \tau)) = \tau$. If $p = (T, \tau)$ and $\sigma \supseteq \tau$, we use p_σ to denote the condition (T_σ, σ).

The complexity of this notion of forcing depends on the representation or indexing used for the infinite recursive binary trees. While,at one end we could use the recursive listing of Exercise 11.1.7, it would then be more difficult to describe various operations on trees that determine subtrees in the natural sense but do not obviously produce an index of the type required. In this case we would also want to define the subtree relation $T \subseteq S$ in terms of $[T] \subseteq [S]$ which would then be a Π^0_2 relation (Exercise 11.1.16) and so only recursive in $0''$.

ubtreepaths **Exercise 11.1.16** *If e and i are indices for infinite binary recursive trees T and S then the relation $[T] \subseteq [S]$ is Π^0_2, and, in fact,it is Π^0_2 complete.*

At the other end, we can simply use indices for recursive functions that define infinite binary trees. While this set is only recursive in $0''$ (because it takes $0''$ to decide if an index is one for a recursive tree), operations on trees become easy to implement on the indices. On this set of indices, the standard subtree relation $T \subseteq S$ is then Π^0_1 and so recursive in $0'$. We adopt this representation of trees for our notion of forcing. In fact, while the notion of forcing is then only $0''$-recursive, some of what we want to do can be done recursively in $0'$ by analyzing the required density functions. As an example, we have the following Lemma.

Lemma 11.1.17 *There is a density function f for the class $V_n = \{(T,\tau) | \, |\tau| \geq n\}$ of dense sets in $\mathcal{P}$ which is recursive in $0'$.*

Proof. Given $p = (T,\tau) \in P$ and $n \in \mathbb{N}$, Lemma 11.1.6 tells us that we can find a $\sigma \in T$ ($\sigma \supseteq \tau$) of length $m \geq n$ such that T_σ is infinite. Clearly $p_\sigma = (T_\sigma, \sigma) \in P$ and $V(p_\sigma) \geq n$. ∎

lowb **Theorem 11.1.18 (Low Basis Theorem)** *If T is a recursive infinite binary tree then it has a low path, i.e. there is a $G \in [T]$ with $G' \equiv_T 0'$. Equivalently, if $\mathcal{C}$ is a nonempty Π^0_1 class, then it has a low member. Moreover, we can compute such a path uniformly recursively in $0'$ and the index for T or the class.*

Proof. As usual we want to show that the sets of conditions deciding the jump ($D_n = \{p | \Phi_n^{V(p)}(n) \downarrow$ or $(\forall q \leq_{\mathcal{P}} p)(\Phi_n^{V(q)}(n) \uparrow)\}$) are dense and provide a density function $f \leq_T 0'$ that also tells us in which way $f(p,n)$ is in D_n. By Lemma 6.1.25 starting with condition $p_0 = (T, \emptyset)$ we can meet these sets as well as the V_n by a generic sequence recursive in $0'$ and so construct a $G \in [T]$ with $G' \equiv_T 0'$ as required.

Given an $p = (T,\tau) \in P$ and an n, we cannot use our usual strategy of asking for a $\sigma \in T$ ($\sigma \supseteq \tau$) such that $\Phi_n^\sigma(n) \downarrow$ and then taking say p_σ as $f(p,n)$ because T_σ may be finite. Instead we ask if $\hat{T} = \{\sigma \in T | \Phi_n^\sigma(n) \uparrow\}$ is infinite. This question can be answered by $0'$ by Lemma 11.1.6. If so, we let $f(p,n) = (\hat{T}, \tau)$ and note that we have satisfied the second clause of the definition of D_n as well as guaranteed that $\Phi_n^G(n) \uparrow$ for every $G \in [\hat{T}]$ including, of course, the generic G we are constructing. If not, then clearly there is a $k \geq |\tau|$ such that $\Phi_n^\sigma(n)\downarrow$ for every $\sigma \in T$ of length k. T_σ must be infinite for one of these σ as T is infinite. Again by Lemma fin0' 11.1.6, $0'$ can find such a σ and we then set $f(p,n) = p_\sigma$. In this case, it is clear that we have satisfied the first clause of D_n and $\Phi_n^G(n)\downarrow$ for every $G \in [T_\sigma]$.

The assertion about members of the corresponding Π_1^0 classes as well as the uniformity claim in the theorem are now immediate. ■

Note that we cannot make a similar improvement to Proposition redegree 11.1.15. Any element of DNR_2, C_{PA} or $S(V_0,V_1)$ of r.e. degree has degree $0'$.??

We next give a different answer to how simple a path we can construct on an arbitrary infinite recursive binary tree. Now the notion of simplicity is specified in terms of domination properties.

0'domb

Theorem 11.1.19 (0′-dominated Basis Theorem) *If T is an infinite recursive binary tree, then there is an $G \in [T]$ such that every $f \leq_T G$ is dominated by some recursive function.*

Proof. We use the same notion of forcing with new dense sets. In place of the D_n we have $E_n = \{(T,\tau)|(\exists x)(\forall \sigma \in T)(\Phi_n^\sigma(x)\uparrow$ or $(\forall x)(\exists k)(\forall \sigma \in T)_{|\sigma|=k}(\Phi_n^\sigma(x)\downarrow)\}$. To see that the E_n are dense consider any condition $p = (T,\tau)$. If there is an x such that $S = \{\sigma \in T | \Phi_n^\sigma(x)\uparrow\}$ is infinite then choose such an x and S. The desired extension of p in E_n is then (S,τ). Note that in this case, $\Phi_n^G(x)\uparrow$ for any $G \in [S]$. If there is no such x, then, by Lemma fin0' 11.1.6, $p = (T,\tau)$ satisfies the second clause in E_n and is itself the witness to density. Note that in this case Φ_n^G is total for any generic G. Indeed, we can now also define a recursive function h which dominates Φ_e^G for any $G \in [T]$: To compute $h(x)$ find a k such that $(\forall \sigma \in T)_{|\sigma|=k}(\Phi_n^\sigma(x)\downarrow$. This is a recursive procedure since by our case assumption there is always such a k. Now set $h(x) = \max\{\Phi_n^\sigma(x)|\sigma \in T$ and $|\sigma| = k\} + 1$. This function clearly dominates Φ_n^G for any $G \in [T]$. ■

Exercise 11.1.20 *Show that we may find a G as in Theorem 0'domb 11.1.19 with $G'' \leq_T 0''$.*

We next turn to finding paths in trees which are simple in the sense that they do not compute some given (nonrecursive) set C or, more generally, any of some countable collection C_i of nonrecursive sets.

pi01coneav

Theorem 11.1.21 (Cone Avoidance) *If T is an infinite recursive binary tree and $\{C_i\}$ is a sequence of nonrecursive sets, there is an $A \in [T]$ such that $C_i \nleq_T A$ for all i.*

Proof. We modify the proof of density of the E_n of Theorem 0'domb 11.1.19 to get $E_{n,m}$ that guarantee that $\Phi_n^G \neq C_m$. We let $E_{n,m} = \{(T,\tau)|(\exists x)(\forall \sigma \in T)(\Phi_n^\sigma(x)\uparrow$ or $(\exists x)(\Phi_n^\tau(x)\downarrow\neq C_m(x))$ or $(\forall x)(\exists k)(\forall \sigma_0, \sigma_1 \in T)_{|\sigma_0|=k=|\sigma_1|}(\Phi_n^{\sigma_0}(x)\downarrow = \Phi_n^{\sigma_1}(x)\downarrow)\}$. Given any condition (T,τ) we first extend it to $q = (S,\sigma) \in E_n$. If we satisfy the first clause of E_n we satisfy the same clause in $E_{n,m}$. Otherwise, we satisfy the second clause of E_n. We now ask if there are ρ_0,$\rho_1 \in S$ with $\rho_i \supseteq \sigma$ and an x such that the S_{ρ_i} are infinite and $\Phi_n^{\rho_0}(x) \downarrow \neq \Phi_n^{\rho_1}(x) \downarrow$. If so, we choose $i \in \{0,1\}$ such that $\Phi_n^{\rho_i}(x) \neq C_m(x)$ and take q_{ρ_i} as our extension of q (and so of p) which gets into $E_{n,m}$ by satisfying the second clause. If not, we claim that q itself satisfies the third clause of $E_{n,m}$ and that there is a recursive function h such that $\Phi_n^G = h$ for every $G \in [S]$. As for q satisfying the third clause of $E_{n,m}$, consider any x and note that it already satisfies the second clause of E_n. If there were infinitely many k such there are $\sigma_0, \sigma_1 \in T$ of length k with $\Phi_n^{\sigma_0}(x) \downarrow \neq \Phi_n^{\sigma_1}(x) \downarrow$ then we would have been in the previous case as there would then be infinitely many $\sigma \in T$ with $\Phi_n^{\sigma}(x) \downarrow \neq C_m(x)$. Thus we may define $h(x)$ by finding a k as in the third clause of $E_{n,m}$ and setting $h(x) = \Phi_n^{\sigma}(x)$ for any σ in S of length k. We then have that $\Phi_n^G = h$ for every $G \in [S]$. As C_m is not recursive,$\Phi_n^G \neq C_m$ for any $G \in [S]$ and so we also satisfy the requirements of the theorem. ■

Exercise 11.1.22 *Show that we may construct a G as required in Theorem* pi01coneav *11.1.21 such that $G \leq_T 0'' \oplus (\oplus_i C_i)$ and indeed uniformly.*

Exercise 11.1.23 *For one nonrecursive C instead of a countable set of C_i show that we may construct a G as required in Theorem* pi01coneav *11.1.21 such that $G \leq_T 0''$ (but without the uniformity). Hint: use the following exercise.*

Exercise 11.1.24 *Prove that for any infinite recursive binary tree T there are $G_0, G_1 \in [T]$ such that any $C \leq_T G_0, G_1$ is recursive. Moreover, we may find such G_i with $G_i'' \equiv_T 0''$.*

Exercise 11.1.25 *Nonempty Π^0_1 classes such as DNR_2 that have no recursive member are called* special *Π^0_1 classes. Prove that any such class has $2^{\aleph_0}$ many members.*

Exercise 11.1.26 *Strengthen some of previous theorems producing a path in T with some property to producing $2^{\aleph_0}$ many if T is special.*

11.2 Finitely branching trees

Also trees recursive in A (f). Relativizations.

Finitely branching trees, f-bounded, (recursively bounded) essentially the same as binary (recursive) binary trees relativize results to f.

Given a recursive recursively bounded tree can get recursive binary tree which has same paths up to degree by padding.

The sets of paths through infinitely branching trees $T \subseteq \mathbb{N}^{<\omega}$ correspond to closed sets in Baire space. Even for recursive trees finding paths is much more complicated in this setting. Whether such trees even have paths is a Π^1_1 complete question. As for a basis theorem, one says that if there is a path then there is one recursive in the complete Π^1_1 set $\mathcal{O}$.??

Reference for low basis theorem $\overset{\texttt{lowb}}{\text{11.1.18}}$ and Theorem $\overset{\texttt{pi01coneav}}{\text{11.1.21}}$: Jockusch, Soare " Degrees of Members of Π^0_1 Classes" Pacific J. Math 40(1972) 605-616
Pseudo jump operators: Jockusch, Shore " Pseudo jump operators I: the r.e. case" Trans. Amer. Math. Soc. 275 (1983) 599-609; " Pseudo-jump operators II: Transfinite iterations, hierarchies, and minimal covers" JSL 49 (1984) 1205-1236

??Check what need for definition jump, relativization to $0^{(n)}$, Π^0_n classes.??

Chapter 12

Pseudo-Jump Operators: Defining $\mathcal{A}$

In this chapter we consider some generalizations of the Turing jump and its iterates (the REA operators). In §12.1 , we prove the analog of the Friedberg completeness theorem 5.3.1 for these operators. We will also see (Theorem 12.2.4) that they include the n-r.e .and ω-r.e. operators (Definition 12.2.1) derived from the n-r.e. and ω-r.e. sets of Definitions ?? and 4.3.12. The primary application that drives our interest here is to the operator given by the Sacks minimal degree construction (Theorem 9.3.1) which constructs a set A of minimal degree with $A \leq_{wtt} 0'$ (Corollary ??). As sets $X \leq_{wtt} 0'$ are ω-r.e. (Exercise 4.3.15), relativization gives an operator M taking any set C to $M(C)$ a minimal cover of C which is ω-r.e. in C (Exercises ??). Thus this operator will fall into our new hierarchy of operators. In particular, our version of the completeness theorem (Theorem ??) will prove, for example, that every degree above $\mathbf{0}^{(\omega)}$ is a minimal cover.

We will also prove in §12.3 an analog of the Posner-Robinson join theorem (Theorem 12.3.2) for 1-REA operators (Theorem 12.3.1) and the operators that correspond to ω-r.e. operations such as the Sacks minimal cover (Theorem 12.3.4). This will provide our major application of these operators: a natural definition in $\mathcal{D}$ of $\mathcal{A} = \{\mathbf{x} | \exists n(\mathbf{x} \leq \mathbf{0}^{(n)}\}$, the degrees of sets definable in arithmetic (Theorem 12.4.1). Our constructions here can be seen as variations on Cohen forcing with special attention paid to the precise collection of dense sets that are to be met so as to make them more effective in various ways.

Our definition of $\mathcal{A}$ provides (Theorem 12.4.3) a proof of the failure of the Homogeneity Conjecture (??) for $\mathcal{D}$ along the lines of the refutation of the Homogeneity Conjecture for $\mathcal{D}'$ in Theorem 6.3.14. Combined with Theorem 7.2.5, it also provides a refutation for the elementary equivalence version of the homogeneity conjecture (??). In particular, we describe a sentence φ in the language of partial orderings such that $\mathcal{D}(\geq \mathbf{0}^{(\omega)}) \vDash \varphi$ but $\mathcal{D} \vDash \neg\varphi$ in Theorem 12.4.4.

We close this chapter with the introduction in §12.5 of a new type of forcing (Kumabe-Slaman). We use it to prove versions of the join theorem for other REA operators. One such join theorem then plays a crucial role in the definition of the Turing jump operator in $\mathcal{D}$ in Chapter 4.1.1.

REA **Definition 12.0.1** *For each $e \in \omega$, we define the* pseudojump operator J_e *on sets A: $J_e(A) = A \oplus W_e^A$. Such operators are also called* 1-REA *because the image is r.e. in and above A. The* n-REA operators *are given by iterations of (in general distinct) pseudojump operators. For $\mu = \langle m_0, m_1, \ldots, m_{n-1}\rangle$ we define the n-REA operator J_μ by $J_\mu(A) = J_{m_{n-1}}(J_{m_{n-2}}(\ldots(J_{m_0}(A)\ldots)$. The* ω-REA operators *are each given by a recursive function f such that $J_f(A) = \oplus_{n\in\omega} J_{f\restriction n}(A)$.*

Example 12.0.2 *If $\mu(i)$ is an index for the usual Turing jump for every $i < n = |\mu|$, then $J_\mu(A) \equiv_T A^{(n)}$. Similarly, if $f(n)$ is an index for usual Turing jump operator for every n, then $J_f(A) \equiv_T A^{(\omega)}$.*

Notation 12.0.3 *For uniformity in decoding, we use both $X \oplus Y$ and $\langle X, Y\rangle$ to denote $\{0\}\times X \cup \{1\}\times Y$. Similarly we code $\langle X_0, \ldots, X_n\rangle$ and $X_0 \oplus \cdots \oplus X_n$ as $\cup\{\{i\}\times X_i | i \leq n\}$. This matches the definition of, e.g. $\oplus_{n\in\omega} X_n$ as $\bigcup\{\{n\} \times X_n\} | n \in \omega\}$ as in Notation* genjoin *??.*

REAcomp 12.1 Completeness Theorems for REA operators

We now prove the completeness theorem for pseudojump, i.e. 1-REA, operators along the lines of the proof of the Friedberg completeness theorem (Theorem frcomp 5.3.1). Let us recall the recursive (in a given $C \geq_T 0'$) procedure for that construction of a set A with $A' \equiv_T C$. It proceeded by constructing finite approximations α_n to A. At step n, it first decided $A'(n)$ by asking $0' \leq_T C$ if there is an extension α of the approximation α_n that forces $\Phi_n^\alpha(n)$ to converge. If there is one, we let α be the first such. If not we let α be α_n. In either case we have decided the jump of A at n. Then we code $C(n)$ into A by setting $\alpha_{n+1} = \alpha\hat{\ }C(n)$.

Here, in addition to replacing the Turing jump with an arbitrary pseudojump operator J_e we do the construction inside any given tree T. (In this chapter trees are binary function trees as in Definition binfunctree 9.2.1.) That is, we construct a subtree S of T by, at each node, looking for an extension on T which decides the next value of J_e on the path being built in T. We also code T' into the path. The path $S[C]$ in S (corresponding to a given $C \geq_T 0'$) is then our desired inversion for J_e, i.e. $J_e(S[C]) \equiv_T C$. We also make explicit the results of applying the operator J_e to any path $S[C]$ on the tree by a *labeling U* assigning σ (and so $S(\sigma)$) to a finite sequence $U(\sigma)$ which is intended to be the initial segment of $W_e^{S[C]}$ that we have already decided for any $C \supset \sigma$. (The formal notion generalizing the labeling we do in this construction is given in Definition nlabel 12.1.5 below.) We make these additions to the simple proof of Theorem frcomp 5.3.1 to lead up to our versions of the completeness theorems first for n-REA operators and then ω-REA ones.

tree1comp **Proposition 12.1.1** *Given an index $e \in \omega$ and a tree T we construct a subtree $S = S_e(T)$ of T and a labeling $U = U_e(T)$ of S such that, for every set C*

1. *$S, U \leq_T T'$.*

2. *$J_e(S[C]) = S[C] \oplus U[C]$.*

3. *For each σ, $J_e(S[C]) \upharpoonright |\sigma|$ is the same for all $C \supset \sigma$.*

4. *$J_e(S[C]) \oplus T \equiv_T C \oplus T' \equiv_T S[C] \oplus T'$.*

5. *$S \oplus U \oplus T \equiv_T T'$.*

Proof. Construction: We define $S(\sigma)$ and $U(\sigma)$ by recursion beginning with $S(\emptyset) = T(\emptyset)$ and $U(\emptyset) = \emptyset$. Suppose we have defined them both for σ of length $n \geq 0$ with $S(\sigma) = T(\rho)$. We define them at $\sigma\hat{}i$ (for $i \in \{0,1\}$) as follows. Ask T' if there is a $\tau \supseteq \sigma$ such that $\Phi_e^{T(\tau)}(n)\downarrow$. If so let ρ be the first such extension found in a standard search recursive in T. (In this case, we have forced $n \in W_e^A$, or equivalently $\langle 1, n\rangle \in J_e(A)$, for any $A \in [T]$ with $T(\rho) \subset A$ or equivalently for $A = T[D]$ for any $D \supset \rho$). If not let $\rho = \tau$. (In this case, we have forced $n \notin W_e^A$, or equivalently $\langle 1, n\rangle \notin J_e(A)$, for any $A \in [T]$ extending $T(\rho)$.) We now set $S(\sigma\hat{}i) = T(\rho\hat{}T'(n)\hat{}i)$. In the first case (we forced $n \in W_e^A$), we set $U(\sigma\hat{}i) = U(\sigma)\hat{}1$. In the second case (we forced $n \notin W_e^A$), we set $U(\sigma\hat{}i) = U(\sigma)\hat{}0$.

Verifications:

1. It is clear from the construction that S and U are recursive in T'.

2. It is also clear from the construction that $\cup_{\sigma\subset C} U(\sigma) = U[C] = W_e^{S[C]}$. Thus $J_e(S[C]) = S[C] \oplus U[C]$.

3. Indeed, by construction, $U[C] = W_e^{S[C]}$ is determined up to $|\sigma|$ for all $C \supset \sigma$ while $S[C] \upharpoonright |\sigma|$ is, of course, determined for $C \supset \sigma$ by σ.

4. By (1), it is clear that for any set C, $C \oplus T' \equiv_T S[C] \oplus T'$. By (1) and (2), $J_e(S[C]) \oplus T \leq_T C \oplus T' \equiv_T S[C] \oplus T'$. To see that $J_e(S[C]) \oplus T \geq_T C \oplus T'$, we show, by induction on the length n of σ, that we can determine the initial segments σ of C, the τ such $S(\sigma) = T(\tau)$ and $T'(n-1)$. Suppose we have the σ of length n such that $\sigma \subset C$ and τ such that $T(\tau) = S(\sigma)$. We first ask $J_e(S[C])$ for the value of $W_e^{S[C]}(n)$. If it is 1 then we find (recursively in T) the first $\rho \supseteq \tau$ such that ${}_e^{T(\tau)}(n)\downarrow$. If not, we let $\rho = \tau$. In either case, the construction makes $S(\sigma\hat{}i) = T(\rho\hat{}T'(n)\hat{}i)$.

   Now for $j = 0$ and $j = 1$ the strings $T(\rho\hat{}j)$ are incompatible and so exactly one is an initial segment of $S[C]$. Thus $S[C] \oplus T$ can determine which one is contained in $S[C]$ and so the value of $T'(n)$. Next, we can, in the same way, determine the i such that $T(\rho\hat{}T'(n)\hat{}i) = S(\sigma\hat{}i)$ is an initial segment of $S[C]$ and so $\sigma\hat{}i \subset C$. The corresponding node on T is $\rho\hat{}T'(n)\hat{}i$. This completes the induction and our proof of the Proposition.

5. One direction of the equivalence is (1) above. For the other, combine (2) and (4) with $C = \emptyset$.

This completes the proof of the Proposition. ■

1comp **Theorem 12.1.2 (Completeness Theorem for 1-REA operators)** *If $C \geq_T 0'$ and $e \in \omega$, then there is an A such that $J_e(A) \equiv_T C \equiv_T A \oplus 0'$.*

Proof. In Proposition tree1comp 12.1.1, let $T = I$, the identity tree ($T(\sigma) = \sigma$), and consider the corresponding tree S and labeling U. As I is recursive and $C \geq_T 0'$, Proposition tree1comp 12.1.1(4) says that $J_e(S[C]) \equiv_T C \equiv_T S[C] \oplus 0'$ so $A = S[C]$ is as required in the Theorem. ■

Exercise 12.1.3 *For a tree T define $\Vdash_T$ so as to make our informal usage of forcing and deciding formulas on T agree with our usual definitions of forcing. Hint: There are several possibilities. One takes the conditions to be pairs $\langle \sigma, T(\sigma) \rangle$ for $\sigma \in 2^{<\omega}$ with the valuation given by projection on the second coordinate.*

Exercise 12.1.4 *Prove that with the notation as in Proposition tree1comp 12.1.1, $S \oplus T \equiv_T T'$.*

We now wish to extend the completeness theorem to n-REA operators and then to ω-REA operators. The idea of taking the result one more step from the $S_e(T)$ and $U_e(T)$ provided in Proposition tree1comp 12.1.1 for J_e to something similar for $J_{e_1} \circ J_e$ is to thin out $U_e(T)$ so as to decide J_{e_1} applied to paths on $U_e(T)$ as we decided J_e on T before. This then induces a subtree $S_{e_1e}(T)$ of $S_e(T)$ and a labeling $U_{e_1e}(T)$ of it satisfying the analogous properties for paths on $S_{e_1e}(T)$ that the previous theorem produced for ones on $S_e(T)$. We first give a general definition of labelings and then state a general proposition for working relative to a given tree T and labeling V.

nlabel **Definition 12.1.5** *An n-*labeling *of a tree T is a function $V : 2^{<\omega} \to (2^{<\omega})^{n+1}$ such that for every σ, the first coordinate of $V(\sigma)$ which we denote, as usual by $V(\sigma)_0$, is $T(\sigma)$ and, for every $\sigma \subset \tau$, $V(\sigma) \subset V(\tau)$ by which we mean that $V(\tau)$* extends *$V(\sigma)$ in each coordinate, i.e. for each $i \leq n$, $V(\sigma)_i \subset V(\tau)_i$. (There is an obvious identification of a tree T and a $1-$labeling of T by the identity function, i.e. $V(\sigma) = \langle T(\sigma) \rangle$.)*

- *For an n-labeling V and $C \in 2^{\mathbb{N}}$, $V[C]$ denotes the $(n+1)$-tuple of sets $\langle V[C]_i \rangle$ where $V[C]_i = \cup_{\sigma \in C} V(\sigma)_i$.*

- *If V is an n-labeling of a tree T and U an m-labeling of a tree S for $m > n$, we say that (S, U)* extends *(T, V), $(S, U) \subseteq (T, V)$, if $S \subseteq T$ and, for every σ and τ such that $S(\sigma) = T(\tau)$, $U(\sigma)_i = V(\tau)_i$ for every $i \leq n$. So, in particular, if $S[C] = T[D]$ then $V[D] = \langle U[C]_i | i \leq n \rangle$.*

FuLabel **Notation 12.1.6** *Recall Definition FuTree 9.2.13 of full subtrees of T above σ: $Fu(T, \sigma)(\tau) = T(\sigma\hat{\ }\tau)$. We extend this in the obvious way to n-labelings V of T: $Fu(V, \sigma)(\tau) = V(\sigma\hat{\ }\tau))$ to get the natural n-labeling of $Fu(T, \sigma)$. We denote the pair $(Fu(T, \sigma), Fu(V, \sigma))$ by $Fu((T, V), \sigma)$.*

The idea behind the definition of n-labelings is that, given an index μ for an n-REA operator, we are expecting to build a tree T with an n-labeling V such that for every set C, $V[C] = J_\mu(T[C])$ by making $V[C]_0 = T[C]$ and $V[C]_{i+1} = W_{\mu(i)}^{\langle V[C]_j | j \leq i\rangle}$ for $i < n$. The tree and labeling are to be designed so that, for every $C \geq_T 0^{(n)}$, $V[C] \equiv_T C$ and so our desired inversion of J_μ is $V[C]_0 = T[C]$. Before stating and proving the Proposition that provides the inductive step of this argument we point out some simple but useful facts about the relation between trees and paths on them.

indforcomp **Proposition 12.1.7** *Let $e \in \omega$ and V be an n-labeling of a tree T and $X = T \oplus V$. There is then an extension (S, U) of (T, V) with U an $(n+1)$-labeling of S such that, for any sets C and D with $S[C] = T[D]$*

1. *$S, U \leq_T X'$.*
2. *$J_e(V[D]) = U[C]$.*
3. *$J_e(V[D]) \upharpoonright |\sigma| = U[C] \upharpoonright |\sigma|$ is the same for all $C \supset \sigma$ (and all $D \supset \tau$ where $T(\tau) = S(\sigma)$).*
4. *$J_e(V[D]) \oplus X \equiv_T C \oplus X'$.*
5. *$U \oplus X \equiv_T X'$.*

Proof. The construction and verifications are similar to those of Proposition tree1comp 12.1.1.

Construction: We begin with $S(\emptyset) = T(\emptyset)$ and $U(\emptyset) = V(\emptyset)\hat{}\,\langle\emptyset\rangle$. Suppose we have defined $S(\sigma)$ and $U(\sigma)$ for σ of length m. We ask $V' \leq_T X'$ if there is a $\rho \supseteq \sigma$ such that $\Phi_e^{V(\rho)}(m)\downarrow$. If so, we choose (recursively in $V \leq_T X$) the first such ρ. If not we let $\rho = \sigma$. We now set $S(\sigma\hat{}i) = T(\rho\hat{}X'(m)\hat{}i)$ and $U(\sigma\hat{}i) = V(\rho\hat{}X'(m)\hat{}i)\hat{}\,\langle U(\sigma)_{n+1}\hat{}1\rangle$ in the first case and $U(\sigma\hat{}i) = V(\rho\hat{}X'(m)\hat{}i)\hat{}\,\langle U(\sigma)_{n+1}\hat{}0\rangle$ in the second.

Verifications: Consider any sets C and D such that $S[C] = T[D]$.

1. As in Proposition tree1comp 12.1.1, it is clear from the construction that $S, U \leq_T X'$.
2. It is again clear from the construction that $W_e^{V[D]} = U[C]_{n+1}$ and so $J_e(V[D]) = U[C]$.
3. The actual step by step analysis for the previous conclusion shows again that $J_e(V[D]) \upharpoonright |\sigma| = U[C] \upharpoonright |\sigma|$ is decided by each σ and so is the same for all $C \supset \sigma$ (and all $D \supset \tau$ where $T(\tau) = S(\sigma)$).
4. That $J_e(V[D]) \oplus X \leq_T C \oplus X'$ follows from (1) and (2). For the other direction, we verify that $J_e(V[D]) \oplus X \geq_T C \oplus X'$ essentially as before. We show, by induction on the length m of σ, that we can determine the initial segments σ of C, the τ such $S(\sigma) = T(\tau)$ and $X'(m-1)$. Suppose we have the σ of length m such that $\sigma \subset C$ and τ such that $T(\tau) = S(\sigma)$. We first ask $J_e(V[D])$ for the value of $W_e^{V[D]}(m)$. If it

is 1 then we find (recursively in $V \leq_T X$) the first $\rho \supseteq \tau$ such that that $\Phi_e^{V(\rho)}(n)\downarrow$. If not, we let $\rho = \tau$. In either case, the construction makes $S(\sigma\hat{\ }i) = T(\rho\hat{\ }X'(n)\hat{\ }i)$.

Now for $j = 0$ and $j = 1$ the strings $T(\rho\hat{\ }j)$ are incompatible and so exactly one is an initial segment of $T[D] = S[C]$. Thus $S[C] \oplus T$ can determine which one is contained in $S[C]$ and so the value of $X'(m)$. Next, we can, in the same way, determine the i such that $T(\rho\hat{\ }X'(n)\hat{\ }i) = S(\sigma\hat{\ }i)$ is an initial segment of $S[C]$ and so $\sigma\hat{\ }i \subset C$ and the corresponding node on T is $\rho\hat{\ }X'(n)\hat{\ }i$. This completes the induction and our proof of (4).

5. That $U \leq_T X'$ is (1). Let D be such that $V[D] = U[\emptyset] \leq_T U$. Now apply (2) and (4) to see that $U[\emptyset] \oplus X \geq_T X'$.

This completes the proof of this Proposition. ■

ncompunif **Remark 12.1.8** *The proof of Proposition* indforcomp *12.1.7 shows that its conclusions hold with all possible uniformities. To specify some of them precisely, we note that there are recursive functions $g, h, k, \hat{k}, l$ such that for every $e \in \omega$ and every tree T with n-labeling V and $X = T \oplus V$.*

1. $\Phi_{g(e)}^{X'} = \mathcal{S}_e(X)$, $\Phi_{h(e)}^{X'} = \mathcal{U}_e(X)$ *which is an $(n+1)$-labeling of the tree $\mathcal{S}_e(X)$ and $(\mathcal{S}_e(X), \mathcal{U}_e(X))$ is an extension of (T, V).*

2. *For every sets C and D with $\mathcal{S}_e(X)[C] = T[D]$, $J_e(V[D]) = \mathcal{U}_e(X)[C]$, $\Phi_{k(e)}^{J_e(V[D])\oplus X} = C \oplus X'$ and $\Phi_{\hat{k}(e)}^{C\oplus X'} = J_e(V[D]) \oplus X$.*

3. *Given σ, the value of $J_e(\mathcal{S}_e(X)[C])(z) = \mathcal{U}_e(X)(z)$ for any $z < |\sigma|$ is the same for all $C \supset \sigma$ and is given by $\Phi_l^{X'}(z, \sigma)$.*

4. $\Phi_{f(e)}^{\mathcal{U}_e(X)\oplus X} = X'$.

Also note that the trees S and U of Proposition tree1comp *12.1.1 are just the $\mathcal{S}_e(X)$ and $(\mathcal{U}_e(X))_1$ of this Proposition with $V = T$.*

We now prove a version of Proposition indforcomp 12.1.7 for n-REA operators by induction using that Proposition as the inductive step along with the uniformities provided by Remark ncompunif 12.1.8.

reenreacomp **Proposition 12.1.9** *For any n-REA operator J_μ and tree T there is a tree T_n with an n-labeling V_n with the following properties:*

1. *Both T_n and V_n are, given μ, uniformly recursive in $T^{(n)}$.*

2. *For every set C, $C \oplus T^{(n)} \equiv_T T_n[C] \oplus T^{(n)} \equiv_T J_\mu(T_n[C]) \oplus T = V_n[C] \oplus T$ with all the reductions given uniformly.*

3. *For each σ and $x < |\sigma|$, the value of $J_\mu(T_n[C]) = V_n[C]$ at x is the same for all $C \supset \sigma$.*

4. *$V_n \oplus T \equiv_T T^{(n)}$.*

Proof. Let J_μ be an n-REA operator with $\mu = \langle m_0, m_1, \dots, m_{n-1}\rangle$. We define by induction trees T_i with i-labelings V_i for $i \leq n$. We let $V_0 = T_0 = T$. Given T_i and V_i we set $X_i = T_i \oplus V_i$. Let $T_{i+1} = \mathcal{S}_{m_i}(X_i)$ and $V_{i+1} = \mathcal{U}_{m_i}(X)$. We claim that T_n and V_n are as required in the Proposition.

Let $C_n = C$ and let C_i for $i < n$ be such that $T_{i+1}[C_{i+1}] = T_i[C_i]$. Note that as $T_0 = V_0$, $T_n[C_n] = T_0[C_0] = V_0[C_0]$.

1. It is immediate from Proposition indforcomp 12.1.7(1) that T_i and V_i are uniformly recursive in $T^{(i)}$ by induction with the desired uniformity (relative to $\mu \restriction i$) given by Remark ncompunif 12.1.8(1). Here, we just need the case $i = n$.

2. By (1), $C \oplus T^{(n)} \equiv_T T_n[C] \oplus T^{(n)}$. By Proposition indforcomp 12.1.7(2) and induction on i we see that $J_{\mu\restriction i}(T_0[C_0]) = J_{\mu\restriction i}(T_i[C_i]) = V_i[C_i]$ and so, in particular, $J_\mu(T_n[C]) = V_n[C_n] = V_n[C]$. Finally, (4) of this Proposition (proved independently below) together with Proposition indforcomp 12.1.7(2) and (4) show by induction that $J_{\mu\restriction i+1}(T_i[C_i]) \oplus T = V_{i+1}[C_{i+1}] \oplus T \equiv_T C_{i+1} \oplus T^{(i+1)}$ and so (for $i = n-1$), $V_n[C_n] \oplus T \equiv_T C_n \oplus T^{(n)} = C \oplus T^{(n)}$ as required.

3. Let $\sigma = \sigma_n$ and choose σ_i such that $T_i(\sigma_i) = T_n(\sigma_n)$. Applying Proposition indforcomp 12.1.7(3) inductively (and the fact established in (2) here that $J_{\mu\restriction i}(T_0[C_0]) = J_{\mu\restriction i}(T_i[C_i]) = V_i[C_i]$) we see that $V_i[C_i] \restriction |\sigma_i|$ is the same for all $C \supset \sigma_i$. Again the case that $i = n$ is what we require here.

4. That $V_i \oplus T \equiv_T T^{(i)}$ follows immediately from Proposition indforcomp 12.1.7(5) by induction.

The uniformities desired here hold for all the T_i and V_i (given as indices relative to $T^{(i)}$) by induction using those in Remark ncompunif 12.1.8. ■

The completeness theorem for n-REA operators is now a special case of Proposition treenreacomp 12.1.9. The one for ω-REA operators is given by another induction using Proposition indforcomp 12.1.7 directly.

nreacomp **Theorem 12.1.10 (Completeness Theorem for n-REA operators)** *If J is an n-REA operator and $C \geq 0^{(n)}$, then there is an A such that $J(A) \equiv_T C \equiv_T A \oplus 0^{(n)}$.*

Proof. Let $T = I$ in Proposition treenreacomp 12.1.9. Thus $T^{(n)} \equiv_T 0^{(n)}$ and $T_n[C]$ is the desired set A by Proposition treenreacomp 12.1.9(2). ■

megareacomp **Theorem 12.1.11 (Completeness Theorem for ω-REA operators)** *If J is an ω-REA operator and $C \geq 0^{(\omega)}$, then there is an A such that $C \equiv_T A \oplus 0^{(\omega)} \equiv_T J(A)$.*

Proof. Construction: Let $J = J_f$ be an ω-REA operator and $C \geq_T 0^{(\omega)}$. We construct a sequence of trees T_i with i-labelings V_i such that (T_{i+1}, V_{i+1}) extends (T_i, V_i) so that $\cup T_i(\emptyset)$ is our desired A. We begin with $T_0 = V_0 = I$. Suppose we have (T_i, V_i). We code in the next number in C by moving first to $Fu((\hat{T}_i, \hat{V}_i), C(i))$. We next apply Proposition indforcomp 12.1.7 to this pair for $J_{f(i)}$ to get (T_{i+1}, V_{i+1}). We now let $A = \cup T_i(\emptyset)$.

Verifications: Note that A is on every T_i and choose C_i such that $T_i[C_i] = A$. We also point out that $C_i(0) = C(i)$ as by the construction here and in Proposition indforcomp 12.1.7 $A \supset T_{i+1}(\emptyset) = \hat{T}_i(\emptyset) = T_i(C(i))$.

We now prove various facts about the construction primarily by applying Proposition indforcomp 12.1.7 inductively and exploiting the uniformities described in Remark ncompunif 12.1.8.

1. We begin working toward the first equivalence of the Theorem. We want to show that T_i and V_i are recursive in $0^{(i)}$ with indices given uniformly in each of $A \oplus 0^{(\omega)}$ and C: We begin with $T_0 = V_0 = I$. Suppose we have (T_i, V_i) with indices computed as required from C and $A \oplus 0^{(\omega)}$. The next step in the construction is to form $\hat{T}_i$ and $\hat{V}_i$. It is clear from the construction that indices for them relative to $0^{(i)}$ can be computed from C (indeed we only need $C(i)$). To see that $A \oplus 0^{(\omega)}$ also suffices, note that $T_i(0)$ and $T_i(1)$ are incompatible extensions of $T_i(\emptyset)$ which are computable from $0^{(i)}$ and then A can decide which of them is an initial segment of A and so compute indices for $\hat{T}_i$ and $\hat{V}_i$ relative to $0^{(i)}$. Now Proposition indforcomp 12.1.7(1) and Remark ncompunif 12.1.8(1) tell us that we can recursively compute indices for T_{i+1} and V_{i+1} from $0^{(i+1)}$ from the ones for $\hat{T}_i$ and $\hat{V}_i$ from $0^{(i)}$ and $f(i)$ while f is a fixed recursive function.

   The uniform computation of these indices from C shows that $A = \cup T_i(\emptyset) \leq_T C$ as $0^{(\omega)} \leq_T C$. On the other hand, the uniform computations from $A \oplus 0^{(\omega)}$ show that T_i (and V_i) are uniformly computable from $A \oplus 0^{(\omega)}$. So too then are the C_i. As $C(i) = C_i(0)$, $C \leq_T A \oplus 0^{(\omega)}$ for the first of the two Turing equivalences asserted by the Theorem.

2. We next note that applying Proposition indforcomp 12.1.7(2) inductively starting with $A = T_0[C_0] = V_0[C_0]$ shows us that $J_{f\upharpoonright i}(A) = V_i[C_i]$ for $i \geq 1$. By (1) here we have that V_i and T_i and hence C_i are uniformly recursive in $A \oplus 0^{(\omega)}$ as, then are $J_{f\upharpoonright i}(A) = V_i[C_i]$. Thus $J_f(A) \leq_T A \oplus 0^{(\omega)}$ for one direction of the second Turing equivalence of the Theorem.

3. Finally, as by (2) and Proposition indforcomp 12.1.7(4), $J_{f\upharpoonright i}(A) = V_i[C_i] \equiv_T C_i \oplus 0^{(i)}$, $0^{(i)} \leq_T J_{f\upharpoonright i}(A)$. The required uniformity follows from Remark ncompunif 12.1.8(3). Thus $A \oplus 0^{(\omega)} \leq_T J_f(A)$.

This completes the proof of the ω-REA completeness theorem. ■

Exercise 12.1.12 *Prove a tree version of Theorem omegareacomp 12.1.11, i.e. given an ω-REA operator J build a tree T such that, for every set $C \geq_T 0^{(\omega)}$, $T[C]$ has the properties required of A in the Theorem.*

Exercise 12.1.13 *The previous exercise allows extending the completeness theorem into the transfinite. Formulate the definition and completeness theorem for $\omega+1$-REA operators and prove the theorem.*

12.2 RE Operators and Minimal Covers

In this section, we generalize the n-r.e. and ω-r.e. sets of Definitions $\overset{\texttt{nre}}{\text{??}}$ and $\overset{\texttt{omegare}}{\text{4.3.12}}$ to corresponding operators. We then show that these operators are, up to degree, all n-REA and ω-REA operators, respectively. As a corollary, we conclude that every degree above $0^{(\omega)}$ is a minimal cover.

reop **Definition 12.2.1** *An operator $J : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is an* n-r.e. operator *if there is an index $e \in \omega$ such that for every $A \in 2^{\mathbb{N}}$, $\Phi_e^A(x,s)$ is total; for every x, $\Phi_e^A(x,0)=0$ and there are at most n numbers s such that $\Phi_e^A(x,s) \neq \Phi_e^A(x,s+1)$ and and $A \oplus \lim_s \Phi_e^A(x,s) = J(A)$. The operator J is* ω-r.e. *if, instead of $\Phi_e^A(x,s)$ changing at most n times for each x, there is a recursive function g such that, for every x, there are at most $g(x)$ many numbers s such that $\Phi_e^A(x,s) \neq \Phi_e^A(x,s+1)$.*

1re1rea **Exercise 12.2.2** *Up to degree, the 1-REA and 1-r.e. operators are the same, i.e. for each 1-REA operator J there is a 1-r.e. operator $\hat{J}$ such that $J(A) \equiv_T \hat{J}(A)$ for every A vice versa. Indeed this result is uniform in the indices. For $n > 1$, however, there are n-REA operators which are not n-r.e. ones even up to degree.*

For future notational simplifications, we note that, up to degree, the function g in the definition of ω-r.e. operators can be taken to be the identity function.

idfunc **Proposition 12.2.3** *For every ω-r.e. operator $\hat{J}$ with witness a recursive g there is (uniformly in the indices for $\hat{J}$ and g) another one J with the identity function as its witness such that, for every $A \in \omega$, $\hat{J}(A) \equiv_T J(A)$.*

Proof. Clearly we may assume that g is increasing by replacing it with $n \longmapsto \Sigma_{i \leq n} g(n)$. Suppose now that $\hat{J}$ is determined by index $\hat{e}$. We define J with index e such that, for every A, x and s, $\Phi_e^A(x,s) = 0$ for $x \in [0, g(0))$ and for $i > 0$ and $x \in [g(i), g(i+1)$, $\Phi_e^A(x,s) = \Phi_{\hat{e}}^A(i,s)$. As $\Phi_{\hat{e}}^A(i,s)$ changes at most $g(i)$ many times, it is clear that $\Phi_e^A(x,s)$ changes at most x many times for each x as required for J to be ω-r.e. It is also clear that for $x > g(0)$, $J(A)(\langle 1, x\rangle) = \lim_s \Phi_e^A(x,s) = \lim_s \Phi_{\hat{e}}^A(i,s) = \hat{J}(\langle 1, i\rangle)$ where is i such that $x \in [g(i), g(i+1))$. As g is recursive, $J(A) \equiv_T \hat{J}(A)$ as well. (As $J(A)(x) = 0$ for $x < g(0)$, it is also clear that the reductions between the two operators are given uniformly in the indices for $\hat{J}$ and g.) ■

As we are only interested in ω-r.e. operators up to degree, we assume from now on that the witness function g required in the definition is always the identity function.

rereaop **Theorem 12.2.4** *For every n-r.e. or ω-r.e. operator $\hat{J}$ there is an n-REA or ω-REA one J, respectively, such that, for every $A \in 2^{\mathbb{N}}$, $\hat{J}(A) \equiv_T J(A)$ and indeed the indices for the REA operator and the required Turing reductions can be found (uniformly) recursively in the ones for the r.e. operator. Moreover, in the n-r.e. case we have $\langle 1, x\rangle \in \hat{J}(A) \Leftrightarrow \langle n, \langle x, 0\rangle\rangle \notin J(A)$ and in the ω-r.e. case we have $\langle 1, x\rangle \in \hat{J}(A) \Leftrightarrow \langle x, \langle x, 0\rangle\rangle \notin J_{f\restriction x}(A)$ where f is as in the definition of an ω-REA operator. (Of course, $\langle 0, x\rangle \in \hat{J}(A) \Leftrightarrow x \in A \Leftrightarrow \langle 0, x\rangle \in J(A)$.)*

Proof. Suppose $\hat{J}$ is an n-r.e. operator with index $\hat{e}$. For $i < n$, We define operators E_i:

$E_i(A) = \{\langle x, s\rangle \mid$ the approximation $\Phi_e^A(x, t)$ has changed from its previous value for the $(n - i - 1)$th time at s and $\Phi_e^A(x, s) \neq \hat{J}(A)(x)\}$.

It is clear that, for each i, $E_i(A) \leq_T \hat{J}(A)$ (uniformly) as $A = \hat{J}(A)^{[0]}$ by definition.

We next claim that, for $i < n$, $E_i(A)$ is (uniformly) r.e. in $E_{i-1}(A)$ (where $E_{-1}(A) = A$). For $i = 0$, note that as $\Phi_e^A(x, t)$ changes at most n times, $\langle x, s\rangle \in E_0(A) \Leftrightarrow$ the approximation $\Phi_e^A(x, t)$ has changed from its previous value for the $(n - 1)$th time at s & $\exists u > s(\Phi_e^A(x, u) \neq \Phi_e^A(x, s))$. For $0 < i < n$, proceed by induction and note that $\langle x, s\rangle \in E_i \Leftrightarrow$ the approximation $\Phi_e^A(x, t)$ has changed from its previous value for the $(n - i - 1)$th time at s & $(\exists u > s)$(the approximation $\Phi_e^A(x, t)$ has changed from its previous value for the $(n - i)$th time at u and $\langle x, u\rangle \notin E_{i-1}$).

Thus there is an n-REA operator J_μ where we define $\mu(i)$ for $i < n$ by induction starting with $J_{\mu(0)}(A) = A \oplus E_{n-1}(A)$ and progressing by making $J_{\mu(i)}(J_{\mu\restriction i}(A)) = J_{\mu\restriction i}(A) \oplus E_{n-i-1}(A)$. All that remains is to show that $\hat{J}(A) \leq_T J_\mu(A)$. In fact, $x \in \hat{J}(A) \Leftrightarrow \langle x, 0\rangle \notin E_{n-1}^A$ as $\Phi_{\hat{e}}^A(x, 0) = 0$ and 0 is the $n - (n - 1) - 1 (= 0)$th time $\Phi_{\hat{e}}^A(x, s)$ has changed for every x. Thus $x \in \hat{J}(A) \Leftrightarrow \langle\langle x, 0\rangle, n\rangle \notin J_\mu(A)$.

It is clear that all the required indices for the operators and Turing reductions are given uniformly in $\hat{e}$ by this construction and the proof of its correctness.

Now let $\hat{J}$ be an ω-r.e. operator with index $\hat{e}$. We can now preform essentially the same procedure for all n. Let $E_i(A) = \{\langle n, s\rangle \mid$ the approximation $\Phi_e^A(n, t)$ has changed from its previous value for the $(n - i - 1)$th time at s and $\Phi_e^A(n, s) \neq \hat{J}(A)(n)\}$. The same arguments as above show that, for $i \geq 0$, $E_{i+1}(A)$ is uniformly r.e. in $E_i(A)$ and recursive in $\hat{J}(A)$. Moreover, $n \in \hat{J}(A) \Leftrightarrow \langle n, 0\rangle \notin E_n(A)$. Thus we may define an ω-REA operator J as in Definition 12.0.1 given by the recursive f such that $J_{f(0)}(A) = A \oplus E_0(A)$ and for $n > 0$, $J_{f(n)}(J_{f\restriction n}(A)) = J_{f\restriction n} \oplus E_n(A)$. The arguments above show that $J(A) \equiv_T \hat{J}(A)$ and, moreover, $x \in \hat{J}(A) \Leftrightarrow \langle x, \langle x, 0\rangle\rangle \notin J_{f\restriction x}(A)$ for every A and x, as required. ■

Corollary 12.2.5 *Every* $\mathbf{c} \geq_T \mathbf{0}^{(\omega)}$ *is minimal cover.*

Proof. As noted at the beginning of this Chapter, there is an ω-r.e. operator M such that for every A, $M(A)$ is a minimal cover of A. (This follows from the uniformities present in the proofs of Theorem 9.3.1, Corollary **??** and Exercise 4.3.15. Theorem 12.2.4 says that, up to degree, there is an ω-REA operator J which also produces minimal covers for every A. Theorem 12.1.11 then provides, for any given $C \geq_T 0^{(\omega)}$, an A for whose degree that of C is a minimal cover.??More details here or when do Sacks minimal cover?? ■

join

12.3 The Join Theorem for ω-r.e. Operators

We begin with the join theorem for 1-REA operators (or equivalently for 1-r.e. operators by Exercise 1re1rea 12.2.2) which will serve as the basic model for our proof of the analogous theorem for ω-r.e. operators.

1join **Theorem 12.3.1** *For every* 1*-REA operator J_e and nonrecursive set X, there is a set A such that $J_e(A) \equiv_T X \oplus 0' \equiv_T A \oplus X$.*

Proof. As in Theorem SW **??**, we may assume that X has no infinite r.e. subsets. (Replace X by the set of all binary strings σ such that $\sigma \subset X$. Any infinite r.e. subset contains an infinite recursive subset which would then compute X for a contradiction.)

We now define initial segments α_n of our desired set A by recursion using $Z = X \oplus 0'$. Assume we have defined α_n. Consider the set $C_n = \{m | (\exists \tau \supseteq \alpha_n \hat{\ } 0^m \hat{\ } 1)(\Phi_e^\tau(0)\downarrow\}$. This set is r.e. and so, if infinite, contains an $m \notin X$. If it is finite, there is certainly an $m \in X$ with $m \notin C_n$. Recursively in Z we can thus search for and find an m such that $m \in C_n \Leftrightarrow m \notin X$. If $m \in C_n$ let τ be the first extension of $\alpha_n \hat{\ } 0^m \hat{\ } 1$ such that $\Phi_e^\tau(0)\downarrow$. If $m \notin C_n$, let τ be $\alpha_n \hat{\ } 0^m \hat{\ } 1$. In either case, set $\alpha_{n+1} = \tau \hat{\ } Z(n)$. Thus $\langle \alpha_n \rangle \leq_T Z$ and so A and $A \oplus X$ are also computable from Z. As we also decided if $n \in W_e^A$ at step n (if and only if $\Phi_e^{\alpha_{n+1}}(n)\downarrow$), $J_e(A) \leq_T Z$.

We now prove that the sequence α_n can be computed from each of $J_e(A)$ and $A \oplus X$. Suppose we have α_n. As $A \leq_T J(A), A \oplus X$, we can find m such that $\alpha_n \hat{\ } 0^m \hat{\ } 1 \subset A$. Now each of X and $J_e(A)$ can tell us if there is a $\tau \supseteq \alpha_n \hat{\ } 0^m \hat{\ } 1$ such that $\Phi_e^\tau(n)\downarrow$ ($\Leftrightarrow m \notin X \Leftrightarrow \langle 1, n \rangle \in J_e(A)$). Given this information we can recursively find the $\tau \supseteq \alpha_n \hat{\ } 0^m \hat{\ } 1$ used at this point of the construction such that $\tau \subset A$. We then know that $\alpha_{n+1} = \tau \hat{\ } Z(n)$ and, of course, A (and hence $J_e(A)$ and $A \oplus X$) can compute this final digit in α_{n+1}. Thus each of $J_e(A)$, $A \oplus X$ can compute α_{n+1} as required. As $Z(n)$ is the last digit of α_{n+1}, both $J_e(A)$ and $A \oplus X$ compute Z. ■

pr **Corollary 12.3.2 (Posner-Robinson join theorem)** *If $0 <_T X \leq_T 0'$ then there is an A such that $0' \equiv_T A' \equiv_T A \oplus X$.*

Proof. Take J_e to be the jump operator in the Theorem. ■

Exercise 12.3.3 *Prove that for every $e \in \omega$, nonrecursive X and $C \geq_T X \oplus 0'$, there is an A such that $J_e(A) \equiv_T C \equiv_T A \oplus X$.*

We now wish to combine the plan of the proof of Theorem 1join 12.3.1 with the constructions in Proposition treenreacomp 12.1.9.

omegajoin **Theorem 12.3.4 (Join Theorem for ω-r.e. Operators)** *If $\hat{J}$ is an ω-r.e. operator, $X \not\leq_T 0^{(n)}$ for every n and $Z \geq_T 0^{(\omega)} \oplus X$, then there is an A such that $\hat{J}(A) \equiv_T Z \equiv_T A \oplus X$.*

Proof. As in Theorem 12.3.1, we may assume that X contains no infinite arithmetic sets. As $\hat{J}$ is an ω-r.e. operator, membership of n in $\hat{J}(A)$ is determined uniformly by n-r.e. operators $\hat{J}_n(A)$ given by $\Phi^A_{e_n}(x,s) = \Phi^A_{\hat{e}}(n,s)$ for every x so that $\hat{J}_n(A)(\langle i,x\rangle) = \hat{J}(A)(\langle 1,n\rangle)$ for every x and $0 < i \leq n$. By Theorem 12.2.4, we (uniformly) have n-REA operators J_{μ_n} such that the $J_{\mu_n}(A)$ are uniformly of the same degree as $\hat{J}_n(A)$. So the $J_{\mu_n}(A)$ are uniformly recursive in A and $\hat{J}(A)(\langle 1,n\rangle)$ for every A. Also note that, as stated in Theorem 12.2.4, $\langle 1,n\rangle \in \hat{J}(A) \Leftrightarrow \langle n,\langle n,0\rangle\rangle \notin J_{\mu_n}(A)$ for every n and A.

Let $s_0 = 0$ and $s_n = \Sigma_{i<n}(i+1) = \frac{1}{2}n(n+1)$ for $n > 0$.

Construction. Recursively in Z we build a nested sequence of trees T_n such that $A = \cup T_n(\emptyset)$ will be our desired set. We begin with $T_0 = I$. Suppose we have $T_n \leq_T 0^{(s_n)}$. Let $T_{n,m} = Fu(T_n, Z(n)\hat{\ }0^m\hat{\ }1)$ which is also recursive in $0^{(s_n)}$ with index computed from that of T_n uniformly in $Z(n)$ and m. Let $S_{n,m}$ be the result of applying Proposition 12.1.9 to $T_{n,m}$ and J_{μ_n}. By Proposition 12.1.9(1), $S_{n,m} \leq_T (0^{(s_n)})^{(n)} = 0^{(s_{n+1})} \leq_T Z$. Let $k_n = \langle n,\langle n,0\rangle\rangle + 1$ and $\hat{T}_{n,m} = Fu(S_{n,m}, 0^{k_n}) \leq_T 0^{(s_{n+1})}$. By Proposition 12.1.9(3), $J_{\mu_n}(\hat{T}_{n,m}[C])(\langle n,\langle n,0\rangle\rangle) = \hat{J}(\hat{T}_{n,m}[C])(\langle 1,n\rangle$ is the same for all $C \supset 0^{k_n}$. Let this value be $j_{n,m}$ By Remark 12.1.8(3) $j_{n,m}$ can be computed uniformly in $0^{(s_{n+1})}$ and m. Thus by our assumption on X, there is an m such that $j_{n,m} = 1 \Leftrightarrow m \notin X$. We choose such an m and let T_{n+1} be $\hat{T}_{n,m}$.

Verifications: Clearly the T_n are nested and A is a path on each of them. As the whole construction is recursive in Z, $A \leq_T Z$ and so $A \oplus X \leq_T Z$. As the construction also shows that $\hat{J}(A)(n)$ is $j_{n,m}$ for the m such that $T_{n+1} = \hat{T}_{n,m}$, we see that $\hat{J}(A) \leq_T Z$ as well.

For the other directions, we show by induction that $Z(n)$, T_n, and the indices for the reductions required to show that $T_n \equiv_T 0^{(s_n)} \equiv_T Fu(T_n,\sigma)$ are uniformly recursive in σ and each of $\hat{J}(A)$ and $A \oplus X$. If we know that $T_n \leq_T \hat{J}(A), A \oplus X$, then using the fact that A lies on T_n we can determine the j and m such that $T_n(j\hat{\ }0^m\hat{\ }1)$ is an initial segment of A. This determines $Z(n) = j$ and the m such that $\hat{T}_{n,m} = T_{n+1}$. For $n = 0$, $T_0 = I$ and $s_0 = 0$ so both T_0 and $0^{(s_0)}$ are recursive. Thus we only have to show that given that we have computed $Z(n)$, m, T_n and the reductions showing (given σ) that $T_n \equiv_T 0^{(s_n)} \equiv_T Fu(T_n,\sigma)$ both $\hat{J}(A)$ and $A \oplus X$ can compute $\hat{T}_{n,m}$ and the reductions showing (given τ) that it is of the same degree as $0^{(s_{n+1})}$ and $Fu(\hat{T}_{n,m},\tau)$. With what we are assuming, we certainly have computed $T_{n,m} = Fu(T_n, Z(n)\hat{\ }0^m\hat{\ }1) \equiv_T 0^{(s_n)}$ and so $\hat{T}_{n,m} = Fu(S_{n,m}, 0^{k_n}) \leq_T S_{n,m} \leq_T (0^{(s_n)})^{(n)} = 0^{(s_{n+1})}$ with the last reduction given by Proposition 12.1.9(1). Now we already know that $J_{\mu_n}(\hat{T}_{n,m}[W])$ is (uniformly) recursive in $\hat{T}_{n,m}[W]$ and $\hat{J}(\hat{T}_{n,m}[W])(n) = j_{n,m}$ for every W. Thus by Proposition 12.1.9(2), $\hat{T}_{n,m}[W]$ $\oplus j_{n,m} \oplus 0^{(s_n)}$ computes $(0^{s_n})^{(n)} = 0^{(s_{n+1})}$ uniformly for every W. If we take $W = \emptyset$, $\hat{T}_{n,m}[W] \leq_T \hat{T}_{n,m}$ and so $\hat{T}_{n,m} \equiv_T 0^{(s_{n+1})}$ as required. For the uniformity needed in our induction, note that $A \in [\hat{T}_{n,m}]$ and both $\hat{J}(A)$ and $A \oplus X$ have computed A and $j_{n,m}$ uniformly and so compute $0^{(s_{n+1})}$ uniformly by Proposition 12.1.9(2).

Thus we have the required computations of $Z(n)$ and $\hat{T}_{n.m} = T_{n+1} \equiv_T 0^{(s_{n+1})}$ from each of $\hat{J}(A)$ and $A \oplus X$. ■

12.4 The Definability of $\mathcal{A}$ and the Failure of Homogeneity

By Theorem jierarchy ??, the sets definable in arithmetic are precisely those recursive in $0^{(n)}$ for some n. We now give a first order definition in $\mathcal{D}$ of the class $\mathcal{A}$ of the degrees of these sets. We then use this definability result to produce failures of the both the Homogeneity Conjecture homconj ?? and its elementary equivalence version (homconjee ??).

defarith **Theorem 12.4.1** *The class $\mathcal{A}$ of Turing degrees is definable in $\mathcal{D}$. Indeed,*

$$\mathcal{A} = \{\mathbf{y}|(\exists \mathbf{x} \geq \mathbf{y})(\forall \mathbf{w})(\mathbf{w} \vee \mathbf{x} \textit{ is not a minimal cover of } \mathbf{w})\}.$$

Proof. Let $\mathcal{C} = \{\mathbf{y}|(\exists \mathbf{x} \geq \mathbf{y})(\forall \mathbf{w})(\mathbf{w} \vee \mathbf{x}$ is not a minimal cover of $\mathbf{w})\}$. That $\mathcal{C} \subseteq \mathcal{A}$ follows from Theorem omegajoin 12.3.4: If $\mathbf{y} \notin \mathcal{A}$ and $\mathbf{x} \geq \mathbf{y}$ then, of course, $\mathbf{x} \notin \mathcal{A}$. Let $\mathbf{w}$ be the degree of the A of Theorem omegajoin 12.3.4 applied to $X \in \mathbf{x}$ and $Z = X \oplus 0^{(\omega)}$ and J the ω-r.e. operator such that $J(A)$ is a minimal cover of A for every A. By the theorem, $\mathbf{w} \vee \mathbf{x}$ is the degree of $J(A)$ and so a minimal cover of $\mathbf{w} = \deg(A)$. Thus $\mathbf{y} \notin \mathcal{C}$ as required.

For the other direction, consider any $\mathbf{y} \in \mathcal{A}$. There is then an n such that $\mathbf{y} \leq_{\mathbf{T}} \mathbf{0}^{(n)}$ (which will serve as the $\mathbf{x}$ required in the definition). So it suffices to show that $\mathbf{w} \vee \mathbf{0}^{(n)}$ is not a minimal cover of $\mathbf{w}$ for any $\mathbf{w}$ and n. Fix $\mathbf{w}$ and proceed by induction on n. Clearly $\mathbf{w} \vee \mathbf{0}$ is not a minimal cover of $\mathbf{w}$. Suppose that there is a least $n > 0$ such that $\mathbf{w} \vee \mathbf{0}^{(n)}$ is a minimal cover of of $\mathbf{w}$. By the minimality of n, $\mathbf{w} \vee \mathbf{0}^{(n-1)} \in [\mathbf{w}, \mathbf{w} \vee \mathbf{0}^{(n)}]$ is not a minimal cover of $\mathbf{w}$. Thus $\mathbf{w} \vee \mathbf{0}^{(n-1)} = \mathbf{w}$. Now note that $\mathbf{0}^{(n)}$ is r.e. in $0^{(n-1)}$ and so also in $\mathbf{w} = \mathbf{w} \vee \mathbf{0}^{(n-1)}$. We now have a contradiction to Corollary recohen 8.2.8 relativized to $\mathbf{w}$ as it shows that no nonrecursive r.e. degree is minimal. ■

Relativizing Theorem defarith 12.4.1 to an arbitrary degree $\mathbf{x}$ gives the definability of the relation $\mathbf{x}$ is arithmetic in $\mathbf{y}$ (Definition defarihtin ??), i.e. $\mathbf{x} \leq_{\mathbf{T}} \mathbf{y}^{(n)}$ for some n.

defarithin **Corollary 12.4.2** *For every* $\mathbf{c}$*, the class* $\mathcal{A}^{\mathbf{c}} = \{\mathbf{y} \geq \mathbf{c}|(\exists \mathbf{x} \geq \mathbf{y})(\forall \mathbf{w} \geq \mathbf{c})(\mathbf{w} \vee \mathbf{x}$ *is not a minimal cover of* $\mathbf{z})\}$ *is the class of degrees arithmetic in and above* $\mathbf{c}$*. Thus the relation* $\mathbf{x} \leq_a \mathbf{y}$*,* $\mathbf{x}$ *is arithmetic in* $\mathbf{y}$*, is definable in* $\mathcal{D}$*.*

Proof. The first assertion is just the relativization of the Theorem to $\mathbf{c}$. Now clearly $\mathbf{x} \leq_a \mathbf{y} \Leftrightarrow \exists \mathbf{u}(\mathbf{u} \in \mathcal{A}^{\mathbf{y}}\ \&\ \mathbf{x} \leq \mathbf{u})$ which, as a relation on $\mathbf{x}$ and $\mathbf{y}$, is definable by the first assertion. ■

nothom **Theorem 12.4.3** *The homogeneity conjecture fails. Indeed, for any degrees* $\mathbf{u}$ *and* $\mathbf{v}$*, if* $\mathcal{D}(\geq \mathbf{u}) \cong \mathcal{D}(\geq \mathbf{v})$ *then* $\mathbf{u} \equiv_a \mathbf{v}$*.*

Proof. Suppose $\mathbf{u}$ is not arithmetic in $\mathbf{v}$ and choose a set $U \in \mathbf{u}$ not arithmetic in any $V \in \mathbf{v}$. As in Definition effsuc ??, let $\mathcal{L}_U$ be an effective successor structure generated by finitely many elements with two additional elements g_0, g_1 such that $n \in U \Leftrightarrow d_n \leq g_0, g_1$. Note that it is clear from its definition that U is arithmetic in the any presentation of $\mathcal{L}_U$ as a partial lattice. Now as $\mathcal{L}_U$ is recursive in U, Theorem 6.3.7 says that we can embed it (as a partial lattice) in $[\mathbf{u}, \mathbf{u}'']$ and so in $\mathcal{A}^{\mathbf{u}}$, the degrees arithmetic in and above $\mathbf{u}$. By Corollary 12.4.2, $\mathcal{A}^{\mathbf{u}}$ is precisely the class of degrees $\mathbf{y}$ such that $\mathcal{D}(\geq \mathbf{u}) \vDash (\exists \mathbf{x} \geq \mathbf{y})(\forall \mathbf{w})(\mathbf{w} \vee \mathbf{x}$ is not a minimal cover of $\mathbf{w})$. If $\mathcal{D}(\geq \mathbf{u})$ and $\mathcal{D}(\geq \mathbf{v})$ were isomorphic, then the isomorphism must take $\mathcal{A}^{\mathbf{u}}$ to $\mathcal{A}^{\mathbf{v}}$ because they have a common definition. Thus $\mathcal{L}_U$ would also be embeddable in $\mathcal{A}^{\mathbf{v}}$. This image is then arithmetic in the image of the generators and so arithmetic in V for the desired contradiction. The argument that $\mathbf{v}$ is arithmetic in $\mathbf{u}$ is symmetric. ■

While this Theorem shows that the homogeneity conjecture fails in the sense that there is a $\mathbf{c}$ (e.g. $\mathbf{0}^{(\omega)}$) such that $\mathcal{D}$ is not isomorphic to $\mathcal{D}(\geq \mathbf{c})$, it does not supply an elementary difference. We can provide one by by using Theorem 7.2.5.

nothomee **Theorem 12.4.4** *$\mathcal{D} \not\equiv \mathcal{D}(\geq \mathbf{0}^{(\omega)})$, i.e. there is a sentence φ in the language of partial orderings such that $\mathcal{D}(\geq \mathbf{0}^{(\omega)}) \vDash \varphi$ but $\mathcal{D} \vDash \neg\varphi$.*

Proof. As $\mathcal{A}$ and $\mathcal{A}^{\mathbf{0}^{(\omega)}}$ are defined in $\mathcal{D}$ and $\mathcal{D}(\geq \mathbf{0}^{(\omega)})$ by the same formula (Corollary 12.4.2), it suffices to show that they are not elementarily equivalent. By Theorem 7.2.5, it then suffices to show that the structures $\mathcal{M}$ and $\mathcal{N}$ of second order arithmetic with set quantifiers ranging over the sets with degrees in $\mathcal{A}$ and $\mathcal{A}^{\mathbf{0}^{(\omega)}}$, respectively, are not elementarily equivalent. This follows from the implicit definability of $0^{(\omega)}$: There is a formula $\psi(X)$ of first order arithmetic with one free set variable X such that the only set X satisfying it is $0^{(\omega)}$. Thus $\mathcal{N} \vDash \exists X \psi(X)$ but $\mathcal{M} \nvDash \exists X \psi(X)$. as required.

Thus we only need to specify $\psi(X)$. It says that $X^{[0]} = \emptyset$ and $\forall n(X^{[n+1]} = (X^{[n]})'$. As we can define Z' in arithmetic uniformly in Z, this formula is clearly equivalent to an arithmetic one with parameter X as required. ■

Remark 12.4.5 *The proof of Theorem 12.4.4 shows that $\mathcal{A} \not\equiv \mathcal{A}^{(0^{\omega})}$. This theorem does not relativize. Indeed, by Borel determinacy (a theorem of ZFC) there is a $\mathbf{c}$ such that $(\forall \mathbf{d} \geq \mathbf{c})(\mathcal{A}^{\mathbf{c}} \equiv \mathcal{A}^{\mathbf{d}})$. Projective determinacy implies that there is a $\mathbf{c}$ such that $(\forall \mathbf{d} \geq \mathbf{c})(\mathcal{D}(\geq \mathbf{c}) \equiv \mathcal{D}(\geq \mathbf{d}))$. Explanations.*

KS

12.5 The Join Theorem for REA operators

Chapter 13

Global Results

homogeneity, definability, automorphisms

Chapter 14

Defining the Turing Jump

Chapter 15

appendix

Appendices

topology

15.1 Trees, Cantor and Baire space; topology; perfect sets

trees of sequences from alphabet (formally identify with subset of $\mathbb{N}$)

binary trees, n-ary trees, finitely branching, f-branching

paths

Cantor space, Baire space

topology, open, closed, perfect sets

perfect trees

function trees

size of continuum Cantor's theorem, $|2^{\mathbb{N}}| = |\mathbb{N}^{\mathbb{N}}| = |[T]|$ for any perfect binary tree.

To text: A nonrecursive construction in mathematics. WKL_0 trees, paths, tree s.t. any path is DNR. forward ref to recursively inseparable sets. completions of PA.

str

15.2 Structures, Orders and Lattices

structure, structure recursive if

p.o., linear, well

lattice, sublattice, usl, lsl, susl, sls

distributive

Boolean algebras

universality issues $\mathbb{Q}$ as example back and forth

locally finite?

partial lattices??

rpretations

15.3 Interpreting Structures and Theories

first and second order logic

Chapter 16

bib

Bibliography

Abraham, U. and Shore, R. A. [1986], Initial segments of the Turing degrees of size $\aleph_1$, *Israel J. Math.* **55**, 1-51.

Cai, M. [2012], Array nonrecursiveness and relative recursive enumerability, *J. of Symb. Logic*, to appear.

Cai, M. and Shore, R. A. [2012], Domination, forcing, array nonrecursiveness and relative recursive enumerability, *J. of Symb. Logic*, to appear.

Cooper, S. B. [1989], Degrees of unsolvability complementary between recursively enumerable degrees, *Ann. Math. Logic* **4**, 31-73.

Dekker, J. C. E. [1954], A theorem on hypersimple sets, *Proc. Am. Math. Soc.* **5**, 791-796.

Feferman, S. [1965] Some applications of the notions of forcing and generic sets, *Fund. Math.* **56,** 325–345.

Feiner, L. [1970], The strong homogeneity conjecture, *J. Symb. Logic* **35**, 375-377.

Friedberg, R. M. [1957], A criterion for completeness of degrees of unsolvability, *J. Symbolic Logic* **22**, 159-160.

Greenberg, N. and Montalbán, A. [2004], Embedding and coding below a 1-generic degree, *Notre Dame J. Formal Logic* **44**, 200-216.

Groszek, M. S. and Slaman, T. A. [1983], Independence results on the global structure of the Turing degrees, *Trans. Am. Math. Soc.* **277**, 579-588.

Hinman, P. G. [1969] Some applications of forcing to hierarchy problems in arithmetic, *Z. Math. Logik Grundlagen Math.* **15**, 341–352.

Jockusch, C. G. jr. [1980], Degrees of generic sets, in *Recursion theory: its generalisation and applications* (Proc. Logic Colloq., Univ. Leeds, Leeds, 1979), *London Math. Soc. Lecture Note Ser.*, **45**, Cambridge Univ. Press, Cambridge-New York, 110–139.

Jockusch, C. G., Jr. and Posner, D. B. [1978], Double jumps of minimal degrees. *J. Symbolic Logic* **43** , 715–724.

Kleene, S. C. and Post, E. L. [1954] The upper semi-lattice of degrees of recursive unsolvability, *Ann. of Math.* (2) **59**, 379–407.

Kurtz, S. A. [1983], Notions of weak genericity, *J. Symbolic Logic* **48** , 764–770.

Lerman, M. [1971], Initial segments of the degrees of unsolvability, *Ann. of Math. (2)* **93**, 365-389.

Lerman, M. [1972], On suborderings of the α-recursively enumerable α-degrees, *Ann. Math. Logic* **4**, 369-392.

Lerman, M. [1983], *Degrees of Unsolvability*, Springer-Verlag, Berlin.

Martin, D. A. [1966], Classes of recursively enumerable sets and degrees of unsolvability, *Z. Math. Logik Grundlagen Math.* **12**, 295–310.

Miller, W. and Martin, D. A. [1968], The degrees of hyperimmune sets. *Z. Math. Logik Grundlagen Math.* **14**, 159–166.

Nerode, A. and Shore, R. A. [1980], Second order logic and first order theories of reducibility orderings in *The Kleene Symposium*, J. Barwise, H. J. Keisler and K. Kunen, eds., North-Holland, Amsterdam, 181-200.

Nerode, A. and Shore, R. A. [1980a], Reducibility orderings: theories, definability and automorphisms, *Ann. of Math. Logic* **18**, 61-89.

Nies, A., Shore, R. A. and Slaman, T. [1998], Interpretability and definability in the recursively enumerable degrees, *Proceedings of the London Mathematical Society* (3) **77**, 241-291.

Odifreddi, P. [1989] and [1999] *Classical recursion theory.* Vols. I and II. *Studies in Logic and the Foundations of Mathematics* **125** and **143**. North-Holland Publishing Co., Amsterdam.

Odifreddi, P. and Shore, R. A. [1991], Global properties of local degree structures, *Bul. U. Mat. Ital.* **7**, 97-120.

Rogers, H. Jr. [1987], *Theory of recursive functions and effective computability.* Second edition. MIT Press, Cambridge, MA.

Sacks, G. E. [1961], On suborderings of degrees of recursive unsolvability, *Z. Math. Logik Grundlagen Math.* **7**, 46-56.

Sacks, G. E. [1963], Recursive enumerability and the jump operator, *Trans. Am. Math. Soc.* **108**, 223-239.

Shoenfield, J. R. [1959], On degrees of unsolvability, *Ann. Math. (2)* **69**, 644-653.

Shoenfield, J. R. [1960], An uncountable set of incomparable degrees, *Proc. Am. Math. Soc.* **11**, 61-62.

Shore, R. A. [1981], The theory of the degrees below $\mathbf{0}'$, *J. London Math. Soc.* (3) **24** (1981), 1-14.

Shore, R. A. [1982], Finitely generated codings and the degrees r.e. in a degree **d**, *Proc. Am. Math. Soc.* **84**, 256-263.

Shore, R. A. [1982a], On homogeneity and definability in the first order theory of the Turing degrees, *J. Symb. Logic* **47,** 8-16.

Shore, R. A. [1988], Defining jump classes in the degrees below $0'$, *Proc. Am. Math. Soc.* **104**, 287-292.

Shore, R. A. [2007], Direct and local definitions of the Turing jump, *J. Math. Logic* **7**, 229-262.

Shore, R. A. [2014], Biinterpretability up to double jump in the degrees below $\mathbf{0}'$, *Proc. of the Am. Math.Soc.* **142**, 351-360.

Shore, R. A. and Slaman, T. A. [1999], Defining the Turing jump, *Math. Research Letters* **6**, 711-722

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

Slaman, T. A. [1991], Degree structures, in *Proc. Int. Cong. Math., Kyoto 1990*, Springer-Verlag, Tokyo, 303-316.

Slaman, T. A.. [2008], Global properties of the Turing degrees and the Turing jump in *Computational prospects of infinity. Part I. Tutorials*, *Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap.* **14**, World Sci. Publ., Hackensack, NJ, 83–101.

Slaman, T. A. and Steel, J. R. [1989], *J. of Symb. Logic* **54**, 160-176.

Slaman, T. A. and Woodin, H. W. [1986], Definability in the Turing degrees, *Illinois J. Math.* **30**, 320-334

Soare, R. I. [1987], *Recursively Enumerable Sets and Degrees*, Springer-Verlag, Berlin.

Spector, C. [1956], On degrees of recursive unsolvability, *Ann. Math. (2)* **64**, 581-592

Thomason, S. K. [1970], Sublattices and initial segments of the degrees of unsolvability, *Can. J. Math.* **3**, 569-581.

Sacks [1966]??