Notes on Π^1_0 classes for Math 661
Fall 2002 Notre Dame University

Reed Solomon

May 24, 2007

1 Definitions and basic examples of Boolean algebras

Definition 1.1. A **partially ordered set** (or **poset** for short) is a pair $\langle X, \leq_X \rangle$ where X is a nonempty set, $\leq_X$ is a binary relation on X with the following properties for all $x, y, z \in X$:

$$x \leq_X x$$
$$(x \leq_X y \wedge y \leq_X x) \rightarrow x = y$$
$$(x \leq_X y \wedge y \leq_X z) \rightarrow x \leq_X z$$

If, in addition, $\langle X, \leq_X \rangle$ satisfies the property $(x \leq_X y \vee y \leq_X x)$ for all x and y, then it is called a **total** or **linear order**.

We frequently drop the subscript X from $\leq_X$ as long as the underlying set is clear from the context. If a partial order has a least element, then we denote this element by 0 and if it has a greatest element, then we denote this element by 1. To be more formal, these elements are defined by the following properties: $\forall x \in X(0 \leq x)$ and $\forall x \in X(x \leq 1)$. Be careful not to confuse least elements with minimal elements or greatest elements with maximal elements. An element $u \in X$ is **minimal** if it is not the case that $\exists x \in X(x < u)$ and an element $v \in X$ is **maximal** if it is not the case that $\exists x \in X(v < x)$.

Definition 1.2. Let X be a poset and $A \subset X$. The element $x \in X$ is an **upper bound** for A if $\forall a \in A(a \leq x)$ and x is a **lower bound** for A if $\forall a \in A(x \leq a)$.

A **chain** in a poset X is a linear ordered subset of X. Recall that Zorn's Lemma states that if every chain in a nonempty poset X has an upper bound, then X has a maximal element.

These notes have been compiled from many sources, none of which are adequately cited. I claim no authorship of the results contained within and I am happy to supply detailed references for the sources of the material contained within these notes. For a breif description of some of the sources, see Section 16.

Definition 1.3. A **lattice** is a poset X such that every pair of distinct elements x and y has a least upper bound, denoted $x \vee y$, and a greatest lower bound, denoted $x \wedge y$. A lattice is called **distributive** if the following properties hold for all $x, y, z \in X$:

$$(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$$
$$(x \wedge y) \vee z = (x \vee z) \wedge (y \vee z)$$

A lattice is called **complemented** if it has a greatest element 1, it has a least element 0 distinct from 1, and it satisfies

$$\forall x \exists y (x \vee y = 1 \;\wedge\; x \wedge y = 0)$$

The element y in this condition is called the **complement** of x.

We can finally state the main definition for the first part of the course. A **Boolean algebra** is a complemented distributive lattice. There are a number of simple properties that you can verify as exercises. For example,

1. both $\vee$ and $\wedge$ are commutative: $x \vee y = y \vee x$ and $x \wedge y = y \wedge x$;
2. both $\vee$ and $\wedge$ are associative: $(x \vee y) \vee z = x \vee (y \vee z)$ and $(x \wedge y) \wedge z = x \wedge (y \wedge z)$;
3. $(x \vee y) \wedge y = y$ and $(x \wedge y) \vee y = y$;
4. $x \wedge 1 = x \vee 0 = x$, $x \vee 1 = 1$, and $x \wedge 0 = 0$;
5. every element has a unique complement, so we can denote the complement of x by $\overline{x}$ or by $c(x)$;
6. $\overline{\overline{x}} = x$.

Although we have presented the definition of a Boolean algebra as a structure $\langle X, \leq \rangle$, we could just as easily have defined it as a structure $\langle X, \vee, \wedge, c, 0, 1 \rangle$. In this case, we would have required that the structure satisfy properties (1)–(3) above as well as the distributivity laws and $c(x) \wedge x = 1$ and $c(x) \vee x = 0$. Typically, we will denote the complement of an element x by $\overline{x}$ rather than $c(x)$. We can define the ordering by $x \leq y \Leftrightarrow x \vee y = y$ or equivalently by $x \leq y \Leftrightarrow x \wedge y = x$.

The theory of Boolean algebras can also be developed in terms of commutative rings. A commutative ring with unit which satisfies $\forall b(b \cdot b = b)$ is called a **Boolean ring**. Let B be a Boolean algebra and define $+_B$ and $\cdot_B$ by

$$a \cdot_B b = a \wedge b$$
$$a +_B b = a \bigtriangleup b = (a \wedge \overline{b}) \vee (\overline{a} \wedge b).$$

$\langle B, +_B, \cdot_B \rangle$ is a Boolean ring. Conversely, if $\langle B, +_B, \cdot_B \rangle$ is a Boolean ring, then the partial order $a \leq b \Leftrightarrow a \cdot_B b = a$ defines a Boolean algebra on the set B.

Before proceeding, we present a number of examples of Boolean algebras. These examples all illustrate some feature of Boolean algebras which we will expand on later.

Example 1.4. Let L be any linear order. The **interval Boolean algebra** of L is denoted by $\text{Int}(L)$ and is the set of all half open intervals $[x, y)$ with $x < y$ in L and ordered by the subset relation.

Example 1.5. Let A be any set. The **power set Boolean algebra** of A is the partial order $\langle \mathcal{P}(A), \subset \rangle$.

Example 1.6. Let T be any topological space. The **Boolean algebra of clopen sets** of T is the set of all subsets of T which are both open and closed, partially ordered by the subset relation.

Example 1.7. The **Boolean algebra of finite and cofinite sets** of $\mathbb{N}$ is the collection of all finite subsets of $\mathbb{N}$ and all cofinite subsets of $\mathbb{N}$ (that is, sets whose complement relative to $\mathbb{N}$ is finite) partially ordered by the subset relation. This Boolean algebra is known as a 1-atom.

Example 1.8. Let $\mathbb{N}$ denote the natural numbers. For any sets $A, B \subset \mathbb{N}$, let $A \triangle B$ denote the symmetric difference of A and B:

$$A \triangle B = (A - B) \cup (B - A).$$

Define the equivalence relation $A \sim B \Leftrightarrow |A \triangle B| < \omega$. Let $\mathbb{N}/\sim$ be partially ordered by $A \leq B \Leftrightarrow |A - B| < \omega$. This structure is a Boolean algebra called the Cantor-Bendixson derivative of $\langle \mathcal{P}(\mathbb{N}), \subset \rangle$.

Example 1.9. Let B be any Boolean algebra and let x be a nonzero element in B. The interval $[0, x] \subset B$ is a Boolean algebra which has least element 0 and greatest element x. The functions $\vee$ and $\wedge$ are simply restricted to this interval and the complement is given by $\overline{y} = x \wedge \overline{y} = x - y$.

2 Filters and quotient algebras

For the rest of these notes, B will denote an arbitrary Boolean algebra. The notions of a filter and an ideal are dual in Boolean algebras, so we can develop either one and get the other for free. Although ideals are more natural in general commutative rings, filters in Boolean algebras have a natural interpretation as logical theories which we will exploit later. Therefore, we develop filters explicitly rather than ideals.

Definition 2.1. A **filter** F in B is a nonempty subset of B such that

$$\forall x, y \in F (x \wedge y \in F)$$
$$\forall x \in F \forall y \in B (x \leq y \rightarrow y \in F).$$

The dual object is called an **ideal** in B. That is, a nonempty subset I of B such that

$$\forall x, y \in I (x \vee y \in I)$$
$$\forall x \in I \forall y \in B (y \leq x \rightarrow y \in I).$$

Notice that for any filter F, $1 \in F$ and for any ideal I, $0 \in I$. For our purposes, we almost always assume our filters and ideals are proper. That is, we assume $F \neq B$ and $I \neq B$, or equivalently, $0 \notin F$ and $1 \notin I$.

We will be interested in when an arbitrary subset A of B can be extended to a (proper) filter. A has the **finite meet property** if the meet of any finite subset of A is not equal to 0. For a finite set $X \subset B$, we denote the meet of the elements of X by $\bigwedge X$ and the join of the elements of X by $\bigvee X$.

Lemma 2.2. *In any Boolean algebra B, $x \wedge \overline{y} = 0$ if and only if $x \leq y$.*

Proof. First, suppose that $x \wedge \overline{y} = 0$. Then,

$$x = x \wedge 1 = x \wedge (y \vee \overline{y}) = (x \wedge y) \vee (x \wedge \overline{y}) = (x \wedge y) \leq y.$$

Second, suppose that $x \leq y$. Then, $x \wedge y = x$ and

$$x \wedge \overline{y} = (x \wedge y) \wedge \overline{y} = x \wedge (y \wedge \overline{y}) = x \wedge 0 = 0.$$

□

Lemma 2.3. *If A has the finite meet property, then for any element $x \in B$, either $A \cup \{x\}$ or $A \cup \{\overline{x}\}$ has the finite meet property.*

Proof. Suppose the lemma is false for A. Then there are finite subsets A_0 and A_1 of A for which $\bigwedge A_0 \wedge x = 0$ and $\bigwedge A_1 \wedge \overline{x} = 0$. However, because $\overline{\overline{x}} = x$, $\bigwedge A_0 \wedge x = 0$ implies $\bigwedge A_0 \leq \overline{x}$. Similarly, $\bigwedge A_1 \leq x$. Therefore, $\bigwedge A_0 \wedge \bigwedge A_1 = 0$, which contradicts the fact that A has the finite meet property. □

In the proof of Lemma 2.3, we used the fact that in any Boolean algebra

$$a \leq x \ \wedge \ b \leq \overline{x} \Rightarrow a \wedge b \leq x \wedge \overline{x} = 0 \Rightarrow a \wedge b = 0.$$

The second implication follows from the fact that 0 is the least element of B. How do we know the first implication is true? We could prove it as a separate lemma (which is a simple exercise), or we could notice that it holds in any power set algebra. That is, for any sets A, B, X

$$A \subset X \wedge B \subset \overline{X} \Rightarrow A \cap B = \emptyset.$$

Later we will show that every Boolean algebra is isomorphic to a subalgebra of a power set algebra. Therefore, any identity which holds in every power set algebra must hold in every Boolean algebra. Of course, to avoid circularity, we should verify all facts without recourse to this method for the present. However, this trick is quite useful in "eye-balling" identities in Boolean algebras.

Lemma 2.4. *A set $A \subset B$ is contained in a proper filter if and only if A has the finite meet property.*

Proof. First, notice that every filter is closed under taking finite meets. Therefore, if A does not have the finite meet property, then any filter containing A must contain 0, and therefore cannot be proper.

Second, assume that A does have the finite meet property. Let A_1 be the finite meet closure of A.

$$A_1 = \{ \bigwedge X \mid X \subset A \wedge |X| < \omega \}$$

Assuming that we understand the meet of a singleton set to be the unique element in that set, it is clear that $A \subset A_1$, that A_1 is closed under taking finite meets, and that $0 \notin A_1$. Let F be the upward closure of A_1.

$$F = \{ x \mid \exists a \in A_1 (a \leq x) \}$$

F is clearly closed upwards and $0 \notin F$. If $x, y \in F$, then there are elements $a_x, a_y \in A_1$ such that $a_x \leq x$ and $a_y \leq y$. Since A_1 is closed under finite meets, $a_x \wedge a_y \in A_1$, and since $a_x \wedge a_y \leq x \wedge y$, $x \wedge y \in F$. Therefore, F is a proper filter containing A. □

It is straightforward to verify that the filter F constructed above is the minimal filter containing A. We will also be concerned with maximum filters containing a given set. Such filters are called ultrafilters.

Definition 2.5. An **ultrafilter** is a proper filter F such that there is no proper filter strictly extending F.

Lemma 2.6. *Let F be a filter in B. F is an ultrafilter if and only if for every $x \in B$, either $x \in F$ or $\overline{x} \in F$, but not both.*

Proof. First, if both x and $\overline{x}$ are in F, then $x \wedge \overline{x} = 0 \in F$ and so F is not proper. Second, if neither x nor $\overline{x}$ is in F, then by Lemma 2.3, either $F \cup \{x\}$ or $F \cup \{\overline{x}\}$ has the finite meet property. Whichever set has the finite meet property can be extended to a proper filter strictly containing F by Lemma 2.4. Therefore, F is not an ultrafilter.

Third, suppose that for each x, either x or $\overline{x}$ is in F, but not both. Let G be any filter extending F. There must be some $x \in G \setminus F$. But, then $\overline{x} \in F$ and so $x, \overline{x} \in G$. Therefore, G is not proper and F must be an ultrafilter. □

Example 2.7. Consider the power set algebra $\mathcal{P}(X)$. For each $A \subset X$, the set $F_A = \{ Y \subset X \mid A \subset Y \}$ is a filter. F_A is called the principal filter generated by A in $\mathcal{P}(X)$.

Definition 2.8. A filter $F \subset B$ is **principal** if there is an element $b \in B$ such that $F = \{x \in B | b \leq x\}$.

We can classify exactly which ultrafilters F in $\mathcal{P}(X)$ are principal.

Lemma 2.9. *A ultrafilter F in $\mathcal{P}(X)$ is principal if and only if there is a finite set $A \in F$.*

Proof. Assume that there is such a finite set $A = \{a_0, \dots, a_n\}$. We claim that the singleton $\{a_i\} \in F$ for some i. If not, then since F is an ultrafilter, the set $X_i = X \setminus \{a_i\}$ is in F for each i. But, the finite intersection of the X_i and A is $\emptyset$ and must be in F. Therefore, F is not proper and hence not an ultrafilter.

For the other direction, assume that F does not contain any finite sets but that F is the principal filter generated by an infinite set A. Let $a \in A$ be any element and consider the sets $\{a\}$ and $X \setminus \{a\}$. Neither of these sets is in F and yet they are complements in $\mathcal{P}(X)$. Therefore, F is not an ultrafilter. □

Corollary 2.10. *An ultrafilter F in $\mathcal{P}(X)$ is principal if and only if it contains a singleton set.*

We are now ready to prove the main result on the existence of ultrafilters.

Theorem 2.11. *Every proper filter F in B can be extended to an ultrafilter.*

Proof. We first prove this result using Zorn's Lemma. Later we will give a more constructive proof for countable Boolean algerbas using Weak König's Lemma. Fix a proper filter F and let $\mathcal{F}$ be the set of all proper filters in B containing F. Notice that $\mathcal{F} \neq \emptyset$ since $F \in \mathcal{F}$. Consider $\mathcal{F}$ as a partially ordered set under inclusion. To apply Zorn's Lemma we only need to show that each chain in this poset has an upper bound.

Let $\{\, G_i \,|\, i \in I \,\}$ be such a chain and let $G = \cup_{i \in I} G_i$. We claim that G is the desired upper bound. It is clear that $F \subset G$. G is proper since $0 \in G$ if and only if $0 \in G_i$ for some i. G is closed upwards since each G_i is closed upwards. If $x, y \in G$, then there are indices i, j such that $x \in G_i$ and $y \in G_j$. Without loss of generality (because the G_k sets form a chain), assume that $G_i \subset G_j$. Then, $x, y \in G_j$, so $x \wedge y \in G_j$ and $x \wedge y \in G$ as required. □

Corollary 2.12. *Every set A with the finite meet property can be extended to an ultrafilter.*

Proof. This is immediate from Lemma 2.4 and Theorem 2.11. □

Corollary 2.13. *Every nonzero element $x \in B$ is in some ultrafilter.*

Proof. The singleton set $\{x\}$ has the finite meet property, so this result follows from the previous corollary. □

Corollary 2.14. *If $x \neq y$ are nonzero elements, then there is an ultrafilter containing one of these elements but not the other.*

Proof. Without loss of generality, assume that $x \not\leq y$. Then, $x \wedge \overline{y} \neq 0$. Therefore, $\{x, \overline{y}\}$ has the finite meet property and can be extended to an ultrafilter. This ultrafilter cannot contain y. □

Example 2.15. We present one more example to show that not all ultrafilters are principal. Let X be an infinite set and let A be the set of all cofinite sets in X. A has the finite intersection property and therefore can be extended to an ultrafilter. However, this ultrafilter does not contain any finite sets and therefore cannot be principal.

A Boolean algebra homomorphism (or just homomorphism) is a map $g : B_1 \to B_2$ between the Boolean algebras B_1 and B_2 which preserves $\wedge$, $\vee$ and complementation. It is straightforward to check that such a map must send 0_{B_0} to 0_{B_1} and 1_{B_0} to 1_{B_1}. A subalgebra of a Boolean algebra B is a nonempty subset A of B which is closed under $\wedge$, $\vee$ and complementation. It is also straightforward to check that a subalgebra must contain 0 and 1.

We will encounter the notions of quotient algebras during this course. Since filters and ideals are dual objects in Boolean algebras, it is not surprising that we can form quotients using either filters or ideals. We consider the definitions for both, since they will both be used during the course.

Let F be a proper filter in B. There are a number of equivalent ways the define the quotient algebra B/F. Two equivalent definitions for the equivalence relation $\sim_F$ are

$$x \sim_F y \Leftrightarrow \exists f \in F(x \wedge f = y \wedge f)$$
$$x \sim_F y \Leftrightarrow x \vee \overline{y} \in F \wedge \overline{x} \vee y \in F.$$

A third possible definition for $\sim_F$ involves defining the relations $\leq_F$ by $x \leq_F y$ if and only if there is an element $f \in F$ such that $x \wedge f \leq y$. Then, $x \sim_F y$ if and only if $x \leq_F y$ and $y \leq_F x$. All of these definitions are equivalent. Under any of these definitions, $\sim_F$ forms a congruence relation and the induced relations on B/F form a Boolean algebra. The natural map which sends x to its equivalence class under $\sim_F$ is a homomorphism from B onto B/F. The elements which are sent to 1 under this map are exactly the elements of F.

Let I be an ideal in B. There are also a number of equivalent ways to define B/I, each of which is dual to one of the definitions above. The simplest definition to work with is

$$x \sim_I y \Leftrightarrow x \triangle y \in I.$$

As above, $\sim_I$ is a congruence relation and the induced structure on B/I gives a Boolean algebra. The elements mapped from B to 0 under the natural homomorphism from B to B/I are exactly the elements of I.

As usual with homomorphisms between algebraic structures, if $g : B_1 \to B_2$ is a homomorphism, then $g(B_1)$ is a subalgebra of B_2. Furthermore, $g^{-1}(1)$ is a filter in B_1 and $g^{-1}(0)$ is an ideal in B_1. Both $B_1/g^{-1}(0)$ and $B_1/g^{-1}(1)$ are isomorphic to $g(B_1)$.

In general for commutative rings, not all prime ideals are maximal. We next show that the analogous statement about filters in Boolean algebras does not hold. A filter F is called **prime** if $x \vee y \in F$ implies either $x \in F$ or $y \in F$.

Lemma 2.16. *A proper filter F is prime if and only if it is an ultrafilter.*

Proof. Suppose F is an ultrafilter, $x \vee y \in F$ and $x \notin F$. Because F is an ultrafilter, $x \notin F$ implies that $\overline{x} \in F$. Because $\overline{x}, x \vee y \in F$, we have

$$\overline{x} \wedge (x \vee y) = (\overline{x} \wedge x) \vee (\overline{x} \wedge y) = \overline{x} \wedge y \in F.$$

But, F is closed upwards, so $y \in F$ as required.

Suppose that F is prime and x is any element of B. Since $x \vee \overline{x} = 1$ and $1 \in F$, we must have either $x \in F$ or $\overline{x} \in F$. However, this condition on F implies F is an ultrafilter. □

3 Lindenbaum algebras

Let $\mathcal{L}$ be a language (either predicate or propositional). We define the **Lindenbaum algebra** $\text{Lind}(\mathcal{L})$ of $\mathcal{L}$. Let $\text{Sent}(\mathcal{L})$ denote the set of sentences in the language $\mathcal{L}$ and define an equivalence relation on this set by $\varphi \sim \psi$ if and only if $\vdash \varphi \leftrightarrow \psi$. The set $\text{Sent}(\mathcal{L})/\sim$ defines the elements of $\text{Lind}(\mathcal{L})$. This equivalence relation is also a congruence relation with respect to the logical operations $\wedge$, $\vee$, $\neg$ and $\rightarrow$. Therefore, we can define the Boolean operations by:

$$[\varphi] \leq [\psi] \Leftrightarrow \vdash \varphi \rightarrow \psi;$$
$$[\varphi] \wedge [\psi] = [\varphi \wedge \psi];$$
$$[\varphi] \vee [\psi] = [\varphi \vee \psi];$$
$$\overline{[\varphi]} = [\neg\varphi].$$

In this representation, the sentences φ for which $[\varphi] = 1$ are exactly the tautologies and the sentences for which $[\psi] = 0$ are exactly the negations of tautologies.

A **theory** T in the language $\mathcal{L}$ is just a set of sentences in this language. If T is closed under logical deduction, then we call T a **closed theory**. Let T be a closed theory. $T \neq \text{Sent}(\mathcal{L})$ if and only if T is consistent. Since T is closed under logical deduction, if $\varphi \in T$ and $\vdash \varphi \rightarrow \psi$, then $\psi \in T$ and if $\varphi, \psi \in T$, then $\varphi \wedge \psi \in T$. In other words, the set of equivalence classes of elements of the closed theory T forms a filter in the Lindenbaum algebra.

Conversely, let F be a filter in $\text{Lind}(\mathcal{L})$ and let T be the set of sentences whose equivalence class lies in F. We claim that T is a closed theory. Since F is nonempty by definition, F contains 1 and hence T contains all the tautologies. Therefore, to see that T is closed under logical deduction, it suffices to check that T is closed under modus ponens; that is, if $\varphi \in T$ and $\varphi \rightarrow \psi \in T$, then $\psi \in T$. Suppose $[\varphi], [\varphi \rightarrow \psi] \in F$. Since F is closed under meets,

$$[\varphi \wedge (\neg\varphi \vee \psi)] = [(\varphi \wedge \neg\varphi) \vee (\varphi \wedge \psi)] \in F.$$

Therefore, $[\varphi \wedge \psi] \in F$. Since F is closed upwards and $[\varphi \wedge \psi] \leq [\psi]$, we have $[\psi] \in F$ and $\psi \in T$ as required.

Therefore, the closed theories in $\mathcal{L}$ correspond exactly to the filters in the Lindenbaum algebra. A closed theory is consistent if and only if 0 is not in the corresponding filter. Therefore, consistent closed theories correspond exactly to proper filters. Furthermore, the complete consistent theories in $\mathcal{L}$ correspond exactly to the ultrafilters in the Lindenbaum algebra. Therefore, in the context of theories, Corollary 2.12 says that any consistent set of sentences can be extended to a complete consistent theory.

These results can all be easily extended to work with provability over a particular (not necessarily closed) theory. That is, if we fix an $\mathcal{L}$-theory T, we could define $\varphi \sim_T \psi$ if and only if $T \vdash \varphi \leftrightarrow \psi$. This relation is also an equivalence relation and the ordering $[\varphi] \leq [\psi]$ if and only if $T \vdash \varphi \rightarrow \psi$ gives a Boolean algebra just as above. This algebra is called the Lindenbaum algebra of the theory T and will be denoted $\text{Lind}_T(\mathcal{L})$ or just Lind_T if the language is clear from the context.

These notions will be useful in the context of computable theories. In particular, if A is a computable set of axioms for a (possibly noncomputable) closed theory T (think about the axioms for Peano arithmetic), then $\text{Lind}_A \cong \text{Lind}_T$.

Finally, to connect these notions to the ideas of quotient algebras, let T be a closed theory and F the associated filter. It is not hard to check that $\mathrm{Lind}_T \cong \mathrm{Lind}(\mathcal{L})/F$.

4 Stone representation theorem

We have referred to the power set algebra $\mathcal{P}(X)$ on several occasions. It is unfortunately not the case that every Boolean algebra has this form. Consider for example the algebra of finite and cofinite subsets of $\mathbb{N}$. This algebra is countable and hence cannot be a power set algebra. However, the Stone representation theorem says that every Boolean algebra is a subalgebra of a power set algebra.

For any Boolean algebra B, let $\mathcal{S}(B)$ be the set of all ultrafilters on B. $\mathcal{S}(B)$ is called the **Stone space** of B. It has a natural topology which we will discuss after proving our first version of the Stone Representation Theorem.

Theorem 4.1. *Any Boolean algebra B is isomorphic to a subalgebra of the power set algebra $\mathcal{P}(\mathcal{S}(B))$.*

Proof. Define the map $g : B \to \mathcal{P}(\mathcal{S}(B))$ by setting

$$g(x) = \{ F \in \mathcal{S}(B) \mid x \in F \}.$$

First, we check that this map is a homomorphism. Because ultrafilters are always prime, an ultrafilter contains $x \vee y$ if and only if it contains x or y. Therefore, $g(x \vee y) = g(x) \cup g(y)$. Because an ultrafilter contains x if and only if it does not contain $\overline{x}$, $g(\overline{x}) = \mathcal{S}(B) \setminus g(x)$. By duality, g must also preserve $\wedge$ and hence it is a homomorphism.

Second, we check that g is one-to-one. We have already seen that if $x \neq y$, then there is an ultrafilter containing one of these elements but not the other. Therefore, $g(x) \neq g(y)$. □

The topology on $\mathcal{S}(B)$ is the topology generated by the sets $g(x)$ for $x \in B$. That is, the basic open sets in $\mathcal{S}(B)$ are precisely the sets of ultrafilters containing a specified element. We next examine such spaces in more detail.

A topological space is a **Boolean space** if it is compact, Hausdorff and has a basis of clopen sets. (That is, a basis of sets which are both closed and open.)

Lemma 4.2. *$\mathcal{S}(B)$ is a Boolean space.*

Proof. Since $g(x) = \mathcal{S}(B) \setminus g(\overline{x})$, each basic open set is also closed. Therefore, $\mathcal{S}(B)$ has a basis of clopen sets.

Suppose that $U, V \in \mathcal{S}(B)$ are distinct ultrafilters. Then, there is an element $u \in U$ with $u \notin V$. Hence $\overline{u} \in V$ and so $U \in g(u)$ and $V \in g(\overline{u})$. Since $g(u) \cap g(\overline{u}) = \emptyset$, these serve as disjoint open sets which separate U and V, which shows that $\mathcal{S}(B)$ is Hausdorff.

To see that $\mathcal{S}(B)$ is compact, it suffices to show that every cover of $\mathcal{S}(B)$ by basic open sets has a finite subcover. For a contradiction, assume that $\{g(x_i) | i \in I\}$ is an infinite cover which does not have a finite subcover. That is, for each finite $I_0 \subset I$, $\cup_{i \in I_0} g(x_i) \neq \mathcal{S}(B)$. Taking complements, we have that

$$\cap_{i \in I_0} g(\overline{x_i}) = g(\bigwedge_{i \in I_0} \overline{x_i}) \neq \emptyset = g(0).$$

Thus, $\bigwedge_{i \in I_0} \overline{x_i} \neq 0$ for any finite set $I_0 \subset I$. This exactly says that $\{\overline{x_i} | i \in I\}$ has the finite intersection property and so can be extended to an ultrafilter U. But, U is not covered by $\{g(x_i) | i \in I\}$ since $\overline{x_i} \in U$ for every $i \in I$. This fact gives the desired contradiction. □

We can now state the second version of the Stone Representation Theorem.

Theorem 4.3. *Any Boolean algebra B is isomorphic to the algebra of clopen subsets of the Boolean space $\mathcal{S}(B)$.*

Proof. From the proof of Theorem 4.1, the image of B under the map g forms a base for the topology on the Boolean space $\mathcal{S}(B)$. Therefore, it suffices to show the following fact: if X is a Boolean space and A is a subalgebra of $\mathcal{P}(X)$ which is base for the topology of X, then A is the algebra of clopen sets in X.

To show that A is contained in the algebra, notice that the elements of A are all open sets since they form a base for the topology. Also, since A is closed under complementation, each element of A is the complement of an open set, and hence is also closed. Therefore, each element of A is clopen.

To show that all clopen sets are contained in A, let Y be such a set. Since A forms a base for the topology, for each $y \in Y$, there is a set $A_y \in A$ such that $y \in A_y$. Without loss of generality (since Y is an open set), we can assume $A_y \subset Y$. Thus $\{A_y | y \in Y\}$ forms an open cover for Y. But, Y is a closed subset of a compact space and hence is compact. Therefore, there is a finite subcover $Y = A_{y_1} \cup \cdots \cup A_{y_n}$. But, A is an algebra, so it is closed under finite unions and $Y \in A$ as required. □

We now have two operations: one to pass from a Boolean algebra B to a Boolean topological space $\mathcal{S}(B)$, and one to pass from a Boolean topological space X to the Boolean algebra of clopen sets. We know that passing from B to $\mathcal{S}(B)$ to the algebra of clopen sets returns an isomorphic Boolean algebra. We next show that passing from a Boolean space X to the Boolean algebra of clopen sets and back to a Boolean space returns a homeomorphic topological space. This is the last version of the Stone Representation Theorem.

Theorem 4.4. *Every Boolean space is homeomorphic to the Stone space of its Boolean algebra of clopen sets.*

Proof. Let X be a Boolean topological space and let B be its Boolean algebra of clopen sets. Let $h : X \to \mathcal{S}(B)$ be defined by

$$h(x) = \{ A \subset X \,|\, A \text{ is clopen} \wedge x \in A \}.$$

Notice that since for each clopen set A, either $x \in A$ or $x \in X \setminus A$ but not both, $h(x)$ is an ultrafilter in B. Therefore, h does map into $\mathcal{S}(B)$.

This map is one-to-one because X is Hausdorff and it has a base of clopen sets. That is, for distinct points x and y in X, there is a clopen set Z such that $x \in Z$ and $y \notin Z$. Therefore $Z \in h(x)$ and $Z \notin h(y)$, so $h(x) \neq h(y)$.

To see that h is onto, let U be an ultrafilter in B. The elements of U are clopen sets in X and U has the finite intersection property. We claim that $\cap\{Z|Z \in U\} \neq \emptyset$. Suppose for a moment that this claim is true and let x be an element of this intersection. Then, $x \in Z$ for all $Z \in U$, so $U \subset h(x)$. But, both U and $h(x)$ are ultrafilters and so $h(x) = U$ as required.

It remains to show that the intersection above is not empty. For a contradiction, assume that $\cap\{Z|Z \in U\} = \emptyset$. Taking complements gives $\cup\{\overline{Z}|Z \in U\} = X$ and hence $\{\overline{Z}|Z \in U\}$ is an open cover of X. Since X is compact, there must be a finite subcover $X = \cup\{\overline{Z}|Z \in U_0\}$ for some finite $U_0 \subset U$. Taking complements again, $\cap\{Z|Z \in U_0\} = \emptyset$. However, U is an ultrafilter and hence has the finite meet property. Therefore, this empty intersection gives the desired contradiction.

Finally, for any $Z \in B$, we claim that

$$\{\, U \in \mathcal{S}(B) \,|\, Z \in U \,\} = \{\, h(x) \,|\, x \in Z \,\}.$$

Notice that this equality finishes the proof since it shows that h maps the base of the topology of X onto the base of the topology for $\mathcal{S}(B)$. Therefore, h is a homeomorphism. To see the containment $\supseteq$, recall that $h(x) = \{A \in B|x \in A\}$. Therefore, if $x \in Z$, then $h(x)$ is an ultrafilter containing Z. To see the containment $\subseteq$, fix an ultrafilter $U \in \mathcal{S}(B)$ with $Z \in U$. Since h is onto, we know $h(x) = U$ for some x. However, since $Z \in U$ and $x \in A$ for all $A \in h(x)$, we have that $x \in Z$ as required. □

5 Interval algebras

In this section, we examine countable Boolean algebras more closely and give another representation theorem for such algebras.

An element $x \in B$ is called an **atom** if $x \neq 0$ and for any $y \in B$, if $y < x$, then $y = 0$. B is called **atomless** if B does not contain any atoms. B is called **atomic** if every nonzero $x \in B$ bounds an atom. Similarly, a nonzero element $x \in B$ is called atomless or atomic just if the algebra $[0, x]$ has that property.

If you are not used to working with atoms in a Boolean algebra, the following theorem provides a good exercise in working through the definitions. It is nothing other than a generalization of Lemma 2.9.

Lemma 5.1. *Let B be a Boolean algebra. An ultrafilter U in B is principal if and only if U contains a finite join of atoms. Furthermore, U contains a finite join of atoms if and only if U contains an atom.*

Proposition 5.2. *If A and B are countable atomless Boolean algebras, then $A \cong B$.*

Proof. By the Stone Representation Theorem, we think of A and B as algebras of sets whenever it helps our intuition during the proof. Our construction of the isomorphism is a back-and-forth argument. We build the isomorphism $g : A \to B$ in finite stages, alternating between satisfying conditions to insure that it is defined on all of A and that it is onto. Let $A_0 = \{0_A, 1_A\}$, $B_0 = \{0_B, 1_B\}$,

$$A \setminus A_0 = \{a_1, a_2, \ldots\},$$
$$B \setminus B_0 = \{b_1, b_2, \ldots\}.$$

We begin the construction at stage 0 by mapping $g_0(0_A) = 0_B$ and $g_0(1_A) = 1_B$. Assume that at the end of stage s, we have defined finite subalgebras A_s of A and B_s of B and we have defined an isomorphism g_s between these algebras.

At stage $s+1$ where $s = 2t$, we make sure that a_t gets into the domain of g_{s+1}. If $a_t \in A_s$, then we let $A_{s+1} = A_s$, $B_{s+1} = B_s$, $g_{s+1} = g_s$ and go to the next stage. Otherwise, let A_{s+1} be the finite subalgebra of A generated by A_s and a_t. (To see that this subalgebra is finite, it is easiest to think of A_s as a finite algebra of sets given by a Venn diagram and a_t adds one new circle to this diagram.) To define g_{s+1} on A_{s+1} it suffices to specify how g_{s+1} is defined on the atoms of A_{s+1} and then extend by taking finite joins. (Again, thinking in terms of algebras of sets is a good idea here. A_s is given by a Venn diagram and a_t adds a new circle to this diagram. Each atomic region in A_s either remains unsplit by this new circle or else it is split into two new atomic regions. These two cases are exactly the ones we consider below.)

Consider each atom x in the finite subalgebra A_s and the effect of adding a_t to A_s on this atom. If $x \wedge a_t = 0$ or $x \wedge \overline{a_t} = 0$, then x remains an atom in A_{s+1} and we can let $g_{s+1}(x) = g_s(x)$.

Otherwise, x is split by a_t into two nonzero pieces $x \wedge a_t$ and $x \wedge \overline{a_t}$. Because $g_s(x)$ is a nonzero element of B and B is atomless, there is a nonzero element $y < g_s(x)$ in B. Fix such a y and notice that y cannot be in B_s since x is an atom in A_s and g_s is an isomorphism from A_s to B_s. Let $z = g_s(x) - y$. Then, $y \vee z = g_s(x)$ and $y \wedge z = 0$. Define $g_{s+1}(x \wedge a_t) = y$ and $g_{s+1}(x \wedge \overline{a_t}) = z$. Since $(x \wedge a_t) \vee (x \wedge \overline{a_t}) = x$, when we extend this map by taking finite unions, we will have $g_{s+1}(x) = g_s(x)$. Also, since $(x \wedge a_t) \wedge (x \wedge \overline{a_t}) = 0$, the new definition of g_{s+1} is consistent with mapping $g_{s+1}(0_A) = 0_B$.

It is clear that the map g_{s+1} on the atoms of A_{s+1} extends naturally to a one-to-one homomorphism into B which extends g_s. Let B_{s+1} be the image of g_{s+1}, so that g_{s+1} is an isomorphism between the finite subalgebras A_{s+1} and B_{s+1}.

At stage $s+1$ where $s = 2t+1$, we make sure that b_t is in the range of g_{s+1} by extending g_s^{-1} in a way analogous to our extension of g_s above. Because $g_s \subset g_{s+1}$ at each stage, we have a limiting map g which is the desired isomorphism from A to B. □

Corollary 5.3. *Let A be a countable Boolean algebra. There is a embedding of A into Int($\mathbb{Q}$).*

Proof. This result is just the "forth" direction of the back-and-forth argument above. □

Because the countable atomless Boolean algebra is determined up to isomorphism, we can use any representation of it that we want. Because different representation are useful is different situations, we mention several different representations for the countable atomless Boolean algebra. The following algebras are all the countable atomless Boolean algebra:

- Int($\mathbb{Q}$);

- the algebra of clopen sets of 2^ω;
- the free Boolean algebra on a countable number of generators a_n, $n \in \omega$;
- $\text{Lind}(\mathcal{L})$ where $\mathcal{L}$ is the propositional language with a countable number of propositional variables A_n, $n \in \omega$.

We can now give our representation theorem for countable Boolean algebras by interval algebras.

Theorem 5.4. *Every countable Boolean algebra B is isomorphic to the interval algebra Int(L) of some linear order $L \subset \mathbb{Q}$.*

Proof. As above, we let $B_0 = \{0_B, 1_B\}$ and $B \setminus B_0 = \{b_1, b_2, \ldots\}$. At stage $s > 0$, we let $B_s = B_0 \cup \{b_1, \ldots, b_s\}$ and B_s^* be the finite subalgebra generated by B_s. We build $L \subset [0,1]$ in stages such that $L_0 = \{0,1\}$ and L_s is a finite linear order which will be a suborder of L. We also build an isomorphism g_s between B_s^* and $\text{Int}(L)$.

At stage 0, $\text{Int}(L)$ has two elements, $\emptyset$ and $[0,1)$. We define $g_0(0_B) = \emptyset$ and $g_0(1_B) = [0,1)$. At the end of stage s, we have the set of atoms $\{a_{s_1}, \ldots, a_{s_n}\}$ in the subalgebra B_s^*, the linear order $L_s \subset \mathbb{Q}$ of size $n+1$ given by $0 = x_{s_0} < x_{s_1} < \cdots < x_{s_n} = 1$, and the map g_s which sends $g_s(a_{s_j}) = [x_{s_{j-1}}, x_{s_j})$. (Notice that these intervals each consist of a single point in L_s.)

At stage $s+1$, check if $b_{s+1} \in B_s^*$. If so, then $B_{s+1}^* = B_s^*$ and we can let $L_{s+1} = L_s$, $g_{s+1} = g_s$ and go to the next stage. Otherwise, we consider the atoms in B_s^* which are split into two pieces in B_{s+1}^* just as in the last proof. (As above, we let $g_{s+1} = g_s$ on the atoms which are not split at this stage.) Let a_i be an atom such that $a_i \wedge b_{s+1} \neq 0_B$ and $a_i \wedge \overline{b_{s+1}} \neq 0_B$. Pick a new point y from the interval (x_{i-1}, x_i) and add y to L_{s+1}. We have now split the interval $[x_{i-1}, x_i)$ into $[x_{i-1}, y) \cup [y, x_i)$. Define g_{s+1} to map $a_i \wedge b_{s+1}$ to one of these new intervals and to map $a_i \wedge \overline{b_{s+1}}$ to the other interval.

We have defined g_{s+1} to map the atoms of B_{s+1}^* to the atoms of $\text{Int}(L_{s+1})$, so we can extend preserving $\vee$ to get an isomorphism between these finite algebras. By the construction, it is clear that $g_s \subset g_{s+1}$. Hence, the limiting map g gives the desired isomorphism from B onto $\text{Int}(L)$. □

One important comment about this representation theorem is that the linear order L is not unique. That is, there are linear orders $L_1 \ncong L_2$ such that $\text{Int}(L_1) \cong \text{Int}(L_2)$.

6 Effectiveness issues

Since this course is intended to address computability issues, this result is a good place to start. We begin by setting up some of the standard terminology from computable model theory. A **countable language** is a (propositional or predicate) language with countably many function symbols $f_0, f_1, \ldots$, countably many relation symbols $R_0, R_1, \ldots$ and countably many constants $a_0, a_1, \ldots$. (Of course they do not need to have these particular names, and there could be only finitely many (or none) of each type of symbol.) A **computable language** is a countable language for which the function, relation and constant symbols form a computable set and there is a computable function assigning the arity to each function and relation symbol.

If $\mathcal{L}$ is a computable language, then a **computable model** $\mathfrak{A}$ for $\mathcal{L}$ is given by a computable set A (the domain of the model) and a uniformly computable interpretation for the function, relation and constant symbols. Equivalently, we could require that the open diagram of $\langle \mathfrak{A}, a \in A \rangle$ is computable. If the language is finite, then it is sufficient to require that each function and relation symbol is interpreted in a computable manner. That is, uniformity is not an issue for finite languages. If the model is supposed to satisfy a set of axioms (such as for a group or field or Boolean algebra), then we require the interpretation of the symbols to satisfy these axioms. If the full diagram of $\mathfrak{A}$ is computable, then we say $\mathfrak{A}$ is a **decidable model**.

Restricting this definition to Boolean algebras, we see that a computable Boolean algebra is given by a computable domain set B together with two constants 0 and 1 and computable functions for $\wedge$, $\vee$ and complementation which satisfy the required axioms.

Frequently, we will start with a particular abstract model (or really an abstract isomorphism type) $\mathfrak{A}$ and want to consider computable models which are isomorphic to $\mathfrak{A}$. A **computable copy** of an abstract model $\mathfrak{A}$ is a computable model which is classically isomorphic to $\mathfrak{A}$. Notice that there can be computable copies of $\mathfrak{A}$ with radically different computable properties. For example, one computable copy might be decidable while a different one is not. A concrete example that is not hard to construct is a computable copy of the linear order type $\langle \omega, \leq \rangle$ in which the successor relation is not computable. This copy differs computationally from the "obvious" presentation in which the successor relation is decidable.

The proof of Proposition 5.2 is effective in the following sense. If B_1 and B_2 are computable atomless Boolean algebras, then the proof of Proposition 5.2 yields a computable isomorphism from B_1 to B_2. Therefore, any two computable copies of the countable atomless Boolean algebra are computably isomorphic and therefore have the same computational properties. (Such structures are called **computably categorical**.) Similarly, the proof of Corollary 5.3 is effective in the sense that if B_1 is any computable Boolean algebra and B_2 is any computable copy of the countable atomless Boolean algebra, then there is a computable embedding from B_1 into B_2. Finally, Theorem 5.4 is effective in the sense that given a computable Boolean algebra B, the proof of this theorem produces a computable linear order L such that B is computably isomorphic to $\text{Int}(L)$.

We will also be concerned with c.e. Boolean algebras. An c.e. Boolean algebra is given by a computable set B, computable functions $\wedge$, $\vee$, and complementation, and an c.e. binary relation $\leq$ on B. For $a, b \in B$, we define $a \sim b$ if and only if $a \leq b$ and $b \leq a$, and we require that $\sim$ is an equivalence relation on which $\wedge$, $\vee$, complementation, and $\leq$ are well defined and such that $B/\sim$ together with these functions and relations forms a Boolean algebra. This Boolean algebra is the c.e. Boolean algebra.

The intuition here is that in an c.e. Boolean algebra, the equality relation is Σ^0_1. We may see that $a \leq b$ and for a long time think that a and b represent different elements of the Boolean algebra, before seeing that $b \leq a$ so that a and b are really two names for the same element of the Boolean algebra.

The most natural example of an c.e. Boolean algebra is given by quotients by c.e. ideals or filters. Let B be a computable Boolean algebra, I an c.e. ideal (meaning $I \subset B$ is an c.e. set), and F an c.e. filter. Then B/I and B/F are both c.e. Boolean algebras. Furthermore, any c.e. Boolean algebra can be realized as $\mathrm{Int}(\mathbb{Q})/F$ for an c.e. filter F in $\mathrm{Int}(\mathbb{Q})$.

To see another natural example of an c.e. Boolean algebra, consider a computable language $\mathcal{L}$ and a computable theory T in this language. The relation $\varphi \sim_T \psi$ from the definition of Lind_T is an c.e. relation. Therefore, Lind_T is an c.e. Boolean algebra. To be more formal, we define an c.e. presentation for this Boolean algebra by letting B be the set of sentences in $\mathcal{L}$, defining meet and join as $\wedge$ and $\vee$, defining the complement of φ to be $\neg\varphi$, and defining $\varphi \leq \psi$ if and only if $T \vdash \varphi \rightarrow \psi$. This presentation gives an c.e. copy of Lind_T and explains the computability theoretic content of the Lindenbaum algebra construction.

7 Tree representations

After the last section, to is natural to ask about the effectiveness of the Stone Representation Theorem for countable algebras. To begin to discuss this issue, we examine an alternate construction of the Boolean space associated to a countable Boolean algebra.

Definition 7.1. A **binary branching tree** T is a subset of $2^{<\omega}$ which is closed under initial segments. A **path** through T is a function $f : \omega \rightarrow 2$ such that for all n, $\langle f(0), \cdots, f(n)\rangle \in T$. We denote the set of all paths through a binary branching tree T by $[T]$.

Let B be a countable Boolean algebra and let $B \setminus \{0, 1\}$ be enumerated as $b_0, b_1, \ldots$. To simplify the notation below, we let $b^1 = b$ and $b^0 = \overline{b}$ for any $b \in B$. We associate a binary branching tree T_B to B and use this tree to represent the Stone space of B. For any string σ of length n,

$$\sigma \in T_B \Leftrightarrow b_0^{\sigma(0)} \wedge b_1^{\sigma(1)} \wedge \cdots \wedge b_{n-1}^{\sigma(n-1)} \neq 0.$$

In other words, $\sigma \in T_B$ if and only if the set $\{b_0^{\sigma(0)}, \ldots, b_{n-1}^{\sigma(n-1)}\}$ has the finite meet property.

Consider any path f through T_B and let $U_f = \{b_n^{f(n)} | n \in \omega\} \cup \{1\}$. We claim that U_f is an ultrafilter of B. First, since our enumeration of the b_n does not contain either 0 or 1, $0 \notin U_f$. Second, suppose $b_n^{f(n)} \leq b_m$. If $f(m) = 0$, then $b_n^{f(n)} \wedge b_m^{f(m)} = 0$ and so $b_0^{f(0)} \wedge \cdots \wedge b_{\max(m,n)}^{f(\max(m,n))} = 0$ which contradicts the fact that f is a path in T. Therefore, $f(m) = 1$ and $b_m \in U_f$. A similar argument shows that if $b_n^{f(n)} \wedge b_m^{f(m)} = b_p$, then $f(p) = 1$ so $b_p \in U_f$. This establishes that U_f is a filter. Since $f(n) \in \{0, 1\}$, either $b_n \in U$ or $\overline{b_n} \in U$ for each b_n. Therefore, U is an ultrafilter.

On the other hand, every ultrafilter U in B can be associated to a path f_U in B. Let $f_U(n) = 1$ if $b_n \in U$ and $f_U(n) = 0$ if $\overline{b_n} \in U$. Notice that f_U is a path in T_B since U has the finite meet property. Since the mappings $U \mapsto f_U$ and $f \mapsto U_f$ are inverse maps, there is a one-to-one correspondence between the ultrafilters in B and the paths through T_B.

To define the appropriate topology on $[T_B]$, consider the full binary branching tree $2^{<\omega}$ and the set $[2^{<\omega}] = 2^\omega$. The basic open sets for the topology on 2^ω are

$$\mathcal{O}_\sigma = \{\, f \in 2^\omega \,|\, \sigma \subset f \,\}$$

for each string $\sigma \in 2^{<\omega}$. Under this topology, 2^ω is a Boolean space. Furthermore, $[T_B]$ is a closed set in 2^ω and so it is also a Boolean space under the subset topology. In fact, $[T_B]$ is homeomorphic to the Stone space of B.

We should also discuss computability issues here. The construction of T_B is effective in the sense that if B is a computable Boolean algebra, then T_B is a computable binary branching tree. Furthermore, the correspondence given above between ultrafilters in B and paths in T_B preserves the Turing degrees of the objects involved. That is, for all ultrafilters U in B, $U \equiv_T f_U$. We have therefore shown the following theorem.

Theorem 7.2. *Let B be a countable Boolean algebra. There is a binary branching tree T_B such that $[T_B]$ is homeomorphic to the Stone space of B. Furthermore, if B is computable, then T_B is computable and the bijection between ultrafilters in B and paths in T_B can be chosen to preserve Turing degree.*

We present the following application of this theorem as our first real taste of the usefulness of tree representations in computable algebra.

Theorem 7.3. *Every computable Boolean algebra has a computable ultrafilter.*

Proof. Fix a computable Boolean algebra B and let T_B be as above. Notice that if σ is a node in T_B of length n, then $b_0^{\sigma(0)} \wedge \cdots \wedge b_{n-1}^{\sigma(n_1)} \neq 0$ and therefore this element can be extended to an ultrafilter. Since the ultrafilters correspond exactly to the paths through T_B, this means that there is a path in T_B that passes through σ. In other words, every node on T_B extends to a path. Therefore, T_B has a computable path f and the associated ultrafilter U_f is a computable ultrafilter. $\square$

In fact, we can say something more here. Consider an c.e. Boolean algebra with underlying set B and equivalence relation $\sim$. We can associate a computable binary branching tree S_B to this Boolean algebra which is slightly different from T_B, but for which $[S_B] = [T_B]$. We build the tree S_B in stages. At stage 0, put the empty string λ into S_B.

At the end of stage s, we will have put all strings of length $\leq s$ into S_B that will ever go into S_B. Assume that $\sigma_0, \ldots \sigma_n$ are the strings of length s in S_B at the end of stage s. At stage $s+1$, we consider each of these strings individually. For σ_i, we consider each string $\tau \subset \sigma_i$. For each such string of length m, run the enumeration of $\leq$ for s many steps and check if $b_0^{\tau(0)} \wedge \cdots \wedge b_{m-1}^{\tau(m-1)} \sim 0$. If so, the do not extend σ_i at stage $s+1$. If not, then add both $\sigma_i * 0$ and $\sigma_i * 1$ to S_B.

The idea here is that we cannot tell right away if σ should be in the tree T_B. So, we take our best guess that it should be in if we have not seen a proper initial segment which generates the 0 element in the c.e. Boolean algebra by stage $|\sigma|$. If we later find out that we were wrong, we terminate all extentions of σ at the stage at which we find out that we are wrong. S_B will have many nodes which cannot be extended to a path, but the nodes which do extend to a path are exactly the nodes that are in T_B. Therefore, $[S_B] = [T_B]$. Furthermore, as above, the natural bijection between ultrafilters in B and paths in S_B preserves Turing degree. We have therefore shown the following theorem.

Theorem 7.4. *For any c.e. Boolean algebra B, there is a computable binary branching tree S_B such that $[S_B]$ is homeomorphic to the Stone space of B. Furthermore, this homeomorphism can be chosen to preserve Turing degrees between the ultrafilters in B and the paths in S_B.*

Unfortunately, we cannot now conclude a theorem like Theorem 7.3 for c.e. Boolean algebras since the computable tree S_B has nodes that do not extend to paths. However, we can relate this result to complete consistent extensions of computable theories. Let T be a consistent computable (not necessarily closed) theory. Lind_T is an c.e. Boolean algebra and therefore there is an associated computable binary branching tree S. The set $[S]$ corresponds exactly to the set of ultrafilters of Lind_T, which in turn correspond exactly to the complete consistent extensions of T. Therefore, we have established the following corollary.

Corollary 7.5. *For any consistent computable theory T, there is a computable binary branching tree S_T and a one-to-one Turing degree preserving bijection between the complete consistent extensions of T and the paths through S_T.*

8 Cantor-Bendixson derivatives

The Cantor-Bendixson derivative is a useful algebraic operation that can be performed on any of the representations of Boolean algebras we have considered so far. It is usually defined in the context of topological spaces.

Definition 8.1. Let X be a topological space. An **isolated point** in X is a point $x \in X$ such that $\{x\}$ is open. The **Cantor-Bendixson** derivative $\mathrm{CB}(X)$ of X is the set of nonisolated points in X.

Consider what this definition says in the context of Stone spaces. Let B be a Boolean algebra and let $\mathcal{S}(B)$ be its Stone space. $\mathrm{CB}(\mathcal{S}(B))$ is formed by removing all isolated ultrafilters from $\mathcal{S}(B)$. By the definition of the topology on $\mathcal{S}(B)$, an ultrafilter U is isolated in $\mathcal{S}(B)$ if and only if there is an element $b \in B$ such that U is the only ultrafilter that contains b.

Lemma 8.2. *An ultrafilter is isolated if and only if it is principal.*

Proof. First, consider an atom $b \in B$ and let F be the principal filter generated by b. (That is, F is the set of all elements above b.) We claim that F is an ultrafilter. Because b is an atom, for all $c \in B$, either $b \leq c$ or $b \wedge c = 0$. But, $b \wedge c = 0$ implies that $b \leq \overline{c}$. So, for any $c \in B$, either $c \in F$ or $\overline{c} \in F$ and hence F is an ultrafilter.

This argument shows that if an atom is in a proper filter than that filter is exactly the principal ultrafilter generated by that atom. Therefore, any ultrafilter containing an atom is isolated. By Lemma 5.1, every principal ultrafilter contains an atom and therefore any principal ultrafilter is isolated.

Conversely, suppose that U is isolated by an element b (that is, it is the unique ultrafilter containing b) but is not principal. By Lemma 5.1, U cannot contain an atom and so b is not an atom. Let c, d be nonzero elements such that $c \wedge d = 0$ and $c \vee d = b$. Since $c \wedge d = 0$, we have that $c \not\leq d$ and $d \not\leq c$. Therefore, the proof of Corollary 2.14 shows that there are ultrafilters U_c and U_d such that $c \in U_c$, $d \notin U_c$ and $c \notin U_d$, $d \in U_d$. Since $c, d \leq b$, both of these ultrafilters contain b. Either $U_c \neq U$ or $U_d \neq U$, and whichever inequality holds contradicts the assumption that U is the only ultrafilter containing b. □

Combining Lemmas 5.1 and 8.2 yields the fact that an ultrafilter is isolated if and only if it contains a finite join of atoms. Therefore, $\mathrm{CB}(\mathcal{S}(B))$ is formed by removing the ultrafilters containing finite joins of atoms from $\mathcal{S}(B)$.

To translate this definition in terms of a binary branching tree T, think of the topological space $[T]$. A path f is isolated in $[T]$ is there is an n such that f is the only path through $f|n$. Therefore, topologically, $\mathrm{CB}([T])$ consists of all the nonisolated paths in $[T]$. In terms of the actual tree T, we define $\mathrm{CB}(T)$ to be the binary branching tree formed by removing all nodes σ from T that have either no paths through them or a unique path through them. Notice that $\mathrm{CB}([T]) = [\mathrm{CB}(T)]$.

To translate this definition in terms of Boolean algebras, consider the Cantor-Bendixson derivative of the Stone space. $\mathrm{CB}(\mathcal{S}(B))$ is formed by removing all the ultrafilters that contain finite joins of atoms. So, to replicate this process in the Boolean algebra, we want to take a quotient that will collapse the finite joins of atoms.

Definition 8.3. Let B be a Boolean algebra. The **Frechet ideal** $\mathcal{F}(B)$ of B is the ideal generated by the atoms of B.

As long as B is not finite, $\mathcal{F}(B)$ is a proper ideal. The Cantor-Bendixson derivative of B is defined to be the quotient algebra $\mathrm{CB}(B) = B/\mathcal{F}(B)$.

Finally, we define the Cantor-Bendixson derivative of a linear order L. Consider the interval algebra $\mathrm{Int}(L)$. In order to mimic the definition in the case of Boolean algebras, we want to collapse all finite joins of atoms in $\mathrm{Int}(L)$. An atom in the interval algebra is given by the half-open interval of a pair of successive elements $[a, b)$. Therefore, the analog of the Frechet ideal is the equivalence relation $a \sim b$ if and only if the interval $[a, b]$ is finite. The Cantor-Bendixson derivative $\mathrm{CB}(\mathrm{Int}(L))$ is the quotient order $L/\sim$. The following lemmas relates these notions and is a good exercise in working out the definitions above.

Lemma 8.4. *If L is a linear order and $B = \mathrm{Int}(L)$, then $\mathrm{CB}(B) \cong \mathrm{Int}(\mathrm{CB}(L))$.*

9 Definitions and basic examples of Π^0_1 classes

We will be concerned with what are sometimes called **recursively bounded** Π^0_1 classes or Π^0_1 classes of sets.

In this section of the notes, we frequently use the variables X, Y, Z to range over subsets of 2^ω. We use the variables f, g, h to range over elements of ω^ω. We use σ, τ to range over strings, either from $\omega^{<\omega}$ or $2^{<\omega}$, which will be clear from the context.

Definition 9.1. A **tree** T is a subset of $\omega^{<\omega}$ that is closed under initial segments. A **finitely branching tree** is a tree T such that there is a function $f \in \omega^\omega$ and all nodes $\sigma \in T$ satisfy $\sigma(n) \leq f(n)$. If this function is computable, then the tree is called a **highly computable tree**. A **binary branching tree** is a subset of $2^{<\omega}$ which is closed under initial segments.

If $f : \omega \to \omega$ is a function, then $f|n$ is the finite sequence $\langle f(0), \dots, f(n-1)\rangle$. Notice that if f happens to map into $\{0, 1\}$, then $f|n$ is a finite binary sequence. If T is any type of tree, we use $[T]$ to denote the set of all functions f such that $\forall n(f|n \in T)$. Any function $f \in [T]$ is called a **path** through T and $[T]$ is called the **set of paths** through T. Notice that the term "path" will always mean "infinite path".

We will be concerned with predicates $R(k, X)$ which are computable or Π^0_1. Recall that there are several equivalent ways to define such predicates. $R(k, X)$ is a **computable** predicate if there is an index e such that $\varphi^X_e(k)$ is defined for all X and all k and $\varphi^X_e(k) = R(k, X)$. Equivalently, $R(k, X)$ is definable by a Δ^0_1 formula in second order arithmetic. $R(k, X)$ is a $\mathbf{\Pi^0_1}$ predicate if there is an index e such that $\varphi^X_e(k, n)$ is defined for all X, k, n and

$$R(k, X) \text{ holds } \Leftrightarrow \forall n(\varphi^X_e(k, n))$$
$$R(k, X) \text{ does not hold } \Leftrightarrow \exists n(\neg\varphi^X_e(k, n)).$$

Equivalently, $R(k, X)$ is definable by a Π^0_1 formula in second order arithmetic. The next lemma gives a useful Normal Form Theorem for Π^0_1 predicates.

Lemma 9.2. *$R(k, X)$ is a Π^0_1 predicate if and only if there is a primitive recursive function f such that*

$$R(k, X) \textit{ holds } \Leftrightarrow \forall n(f(k, n, X|n) = 1)$$
$$R(k, X) \textit{ does not hold } \Leftrightarrow \exists n(f(k, n, X|n) = 0).$$

Proof. Fix an index e as in the definition of a Π^0_1 predicate. Define a computable function f by $f(k, n, \sigma) = 0$ if there is an $m \leq n$ for which $\varphi^\sigma_{e,n}(k, m) \downarrow= 0$, and define $f(k, n, \sigma) = 1$ otherwise. It is not hard to check that f has the required properties. Notice that f is primitive recursive since the Kleene T-predicate is primitive computable and hence for any string σ and any s, the relation $\varphi^\sigma_{e,s}(k, m) = 0$ is a primitive computable relation. □

We begin with a proposition which will yield several definitions for Π^0_1 classes.

Proposition 9.3. *For any class $P \subset \omega^\omega$, the following are equivalent.*

1. *$P = [T]$ for a computable tree $T \subset \omega^{<\omega}$.*
2. *$P = [T]$ for a primitive recursive tree $T \subset \omega^{<\omega}$.*
3. *$P = \{f \in \omega^\omega | \forall n(R(n, f))\}$ for some computable relation R.*
4. *$P = [T]$ for some Π^0_1 tree T.*

Proof. To see (1) ⇒ (2), let $P = [S]$ where $S \subset \omega^{<\omega}$ is a computable tree. Fix an index e for a computable function φ_e such that $\varphi_e(\sigma) = 1$ if $\sigma \in S$ and $\varphi_e(\sigma) = 0$ if $\sigma \notin S$. As usual, we have the approximations $\varphi_{e,s}$ to φ_e. Define a primitive recursive tree T by $\tau \in T$ if and only if

$$\forall n < |\tau|(\neg\varphi_{e,|\tau|}(\tau|n) = 0).$$

Notice that if $\sigma \in S$, then for all $n < |\sigma|$, $\sigma|n \in S$ and hence $\varphi_e(\sigma|n) = 1$. Therefore, $S \subset T$ and so $[S] \subset [T]$. Furthermore, if $f \notin [S]$, then for some n, $f|n \notin S$. Let $\sigma = f|n$ and let s be such that $\varphi_{e,s}(\sigma) \downarrow = 0$. For all τ with $\sigma \subset \tau$ and $|\tau| > s$, we have $\varphi_{e,|\tau|}(\tau|n) = \varphi_{e,|\tau|}(\sigma) = 0$. Hence, all extensions of σ of length longer than s are not in T. Therefore, $f \notin [T]$ as required.

Next we show (2) $\Rightarrow$ (3). Recall that to say $R(n, f)$ is computable means that there is a computable function φ_e such that for all $f \in \omega^\omega$ and all n, $R(n, f) \Leftrightarrow \varphi_e^f(n) = 1$ and $\neg R(n, f) \Leftrightarrow \varphi_e^f(n) = 0$. Define $R(n, f) \Leftrightarrow f|n \in T$. R is clearly computable and has the desired property.

To see (3) $\Rightarrow$ (1), fix an index e such that $R(n, f) \Leftrightarrow \varphi_e^f(n) = 1$ and $\neg R(n, f) \Leftrightarrow \varphi_e^f(n) = 0$. Define a computable tree T by

$$\tau \in T \Leftrightarrow \forall k < |\tau|(\neg\varphi^{\tau}_{e,|\tau|}(k) = 0).$$

Assume that $\forall n R(n, f)$ holds. Then, $\forall n, s(\neg\varphi^{f|s}_{e,s}(n) = 0)$ and so $f \in [T]$. If $\exists n \neg R(n, f)$, then fix such an n. There must be an $s \geq n$ such that $\varphi^{f|s}_{e,s}(n) = 0$. But this implies that $f|s \notin T$ and hence $f \notin [T]$.

It is clear that (1) implies (4), so it only remains to show that (4) implies (1). Let $S \subset \omega^{<\omega}$ be a Π^0_1 tree. There is a computable relation R such that $\sigma \in S$ if and only if $\forall n R(n, \sigma)$. Define T by $\tau \in T$ if and only if $\forall m, n \leq |\tau|(R(m, \tau|n))$. If $\sigma \in S$, then for all $m \leq |\sigma|(\sigma|m \in S)$ and so $\forall n, m \leq |\sigma|(R(n, \sigma|m))$. Therefore, $\sigma \in T$, so $S \subset T$ and $[S] \subset [T]$. If $\sigma \notin S$, then there is an n such that $\neg R(n, \sigma)$. As above, this implies σ has no extensions in $[T]$. □

Definition 9.4. A $\mathbf{\Pi^0_1}$ **class** is a class $P \subset \omega^\omega$ which satisfies any of the conditions from Proposition 9.3. If $P = [T]$ for a finitely branching tree, then P is called a **bounded** $\mathbf{\Pi^0_1}$ **class**, and if T is highly computable, then P is called a **computablely bounded** $\mathbf{\Pi^0_1}$ **class**. A $\mathbf{\Pi^0_1}$ **class of sets** is a class $P \subset 2^\omega$ for which there is a computable tree $T \subset 2^{<\omega}$ with $P = [T]$.

Our main interest will be with Π^0_1 classes of sets. Whenever we refer to a Π^0_1 class of sets $P = [T]$, we assume that T is a computable binary branching tree. Proposition 9.3 can be restated in terms of Π^0_1 classes of sets.

Proposition 9.5. *For any class $P \subset 2^\omega$, the following are equivalent.*

1. *$P = [T]$ for a computable tree $T \subset 2^{<\omega}$.*
2. *$P = [T]$ for a primitive recursive tree $T \subset 2^{<\omega}$.*
3. *$P = \{X \in 2^\omega | \forall n(R(n, X))\}$ for some computable relation R.*
4. *$P = [T]$ for some Π^0_1 tree $T \subset 2^{<\omega}$.*

By Proposition 9.5, there is an effective list of all Π^0_1 classes of sets. Let f_e for $e \in \omega$ effectively enumerate all primitive recursive functions from $2^{<\omega}$ into 2. Define T_e to be a tree such that

$$\sigma \in T_e \Leftrightarrow \forall \tau \subset \sigma(f_e(\tau) = 1).$$

Then the sequence $[T_e]$ enumerates all Π^0_1 classes of sets. When we refer to an **index** for a Π^0_1 class of sets P, we mean an index e such that $P = [T_e]$.

For any finitely branching tree $T \subset \omega^{<\omega}$, the topology on $[T]$ is given by basic open sets

$$\mathcal{O}_\sigma = \{\, f \in [T] \,|\, \sigma \subset f \,\}.$$

Since T is finitely branching, $[T]$ is compact and these basic open sets are also closed. Furthermore, $[T]$ is Hausdorff and hence is a Boolean space. Notice that this topology is exactly the same as the topology we described when we studied the tree representation for the Stone space of a countable Boolean algebra. Recall that we say that $f \in [T]$ is an **isolated path** if there is an n such that f is the unique path in T passing through $f|n$.

Lemma 9.6. *An isolated path $f \in [T]$ in a Π^0_1 class of sets is computable.*

Proof. Let n be such that f is the unique path in T passing through $f|n$. As a finite amount of information, we can assume that we know the values of f for numbers $\leq n$. We define the values of f for numbers greater than this by induction. Assume that $m > n$ and we know $f|m = \langle f(0), \dots, f(m-1)\rangle$. We search the values of nodes in T above $f|m * 0$ and $f|m * 1$ until we find that T is finite above one of these nodes. Notice that it must be finite above one of these nodes since otherwise there would be a path in T passing though $f|m * 0$ and one passing through $f|m * 1$, which contradicts the fact that f is the unique path passing through $f|n \subset f|m$. We set $f(m) = 1 - i$ where i is such that the tree above $f|m * i$ is finite. □

We next show that up to homeomorphism, it does not matter whether we consider Π^0_1 classes of sets or computablely bounded Π^0_1 classes.

Proposition 9.7. *For every highly computable tree T, there is a computable tree $S \subset 2^{<\omega}$ such that the spaces $[T]$ and $[S]$ are computablely homeomorphic.*

Proof. We define a computable mapping ψ from $\omega^{<\omega}$ to $2^{<\omega}$ that sends $\sigma \in \omega^{<\omega}$ to

$$\psi(\sigma) = \langle 0^{\sigma(0)} 1 0^{\sigma(1)} 1 \cdots 0^{\sigma(|\sigma|-1)} 1\rangle.$$

The mapping ψ extends to infinite strings in ω^ω in the obvious way. ψ is not a homeomorphism between ω^ω and 2^ω because ω^ω is not compact while 2^ω is compact. However, let f be a computable function such that for all $\sigma \in T$ and $n < |\sigma|$, $\sigma(n) < f(n)$. If we define the tree

$$S = \{\, \psi(\sigma) * 0^i \,|\, \sigma \in T \wedge i < f(|\sigma|) \,\}$$

then ψ is a homeomorphism between $[T]$ and $[S]$. We leave the details to the reader. □

We conclude this section with several examples of Π^0_1 classes of sets.

Example 9.8. By Proposition 9.5, the class $P_e = \{X | \forall s (\varphi^X_{e,s}(e) \uparrow)\}$ is a Π^0_1 class of sets and there is a computable tree $Q_e \subset 2^{<\omega}$ such that $[Q_e] = P_e$. Similarly, there are computable binary branching trees Q^i_e such that $[Q^i_e] = \{X | \forall s (\varphi^X_{e,s}(i) \uparrow)\}$.

Example 9.9. Let A and B be disjoint computablely enumerable sets. The class $\mathrm{Sep}(A, B)$ of separating sets of A and B is defined to be

$$\mathrm{Sep}(A, B) = \{ C \subset \omega \mid A \subset C \wedge C \cap B = \emptyset \}.$$

$\mathrm{Sep}(A, B)$ is a Π^0_1 class of sets. Π^0_1 classes of this type are called **$\mathbf{\Pi^0_1}$ classes of separating sets**.

Example 9.10. A useful example of a class of separating sets comes from letting $A = \{e|\varphi_e(e) = 0\}$ and $B = \{e|\varphi_e(e) = 1\}$. A set X is in $\mathrm{Sep}(A, B)$ if and only if for all e, $X(e) \neq \varphi_e(e)$. Therefore, this Π^0_1 class of sets gives exactly the DNR_2 functions. (These functions are called "diagonally noncomputable" and the subscript 2 indicates that we are dealing only with $\{0, 1\}$-valued functions.) These sets are extremely useful. In particular, there are two equivalences for when a Turing degree $\mathbf{a}$ is the degree of a DNR_2 set. $\mathbf{a}$ has this property if and only if $\dashv$ is the degree of a complete extension of Peano arithmetic, and also if and only if $\mathbf{a}$ can compute a path through every nonempty Π^0_1 class of sets.

Example 9.11. The union and the intersection of two Π^0_1 classes of sets are Π^0_1 classes of sets. Of course, this example can be extended to finite unions and intersection. Furthermore, taking finite unions and intersections is uniform in the indices for the Π^0_1 classes of sets involved.

Example 9.12. Let P_i for $i \in \omega$ be a computable descending chain of nonempty Π^0_1 classes of sets. That is, $P_{i+1} \subset P_i$ for all i. Then, $P = \cap_{i \in \omega} P_i$ is a nonempty Π^0_1 class of sets. (To see this is so, recall that we can assume that each P_i is given by a computable tree and all we need to show about P is that it is given by a Π^0_1 tree. To see why P is nonempty, notice that it is the intersection of a nested sequence of nonempty closed sets in 2^ω.)

10 Topological basics

Our goal in this section is to examine some of the basic topological facts associated with Π^0_1 classes of sets. Before beginning this discussion, we introduce the standard metric function $d(X, Y)$ on 2^ω. If $X \neq Y$, then let n be the least number such that $X(n) \neq Y(n)$. We define $d(X, Y) = 2^{-n}$.

Lemma 10.1. *The metric $d(X, Y)$ generates the topology we have defined on 2^ω.*

Proof. Fix any set X and any natural number n. The closed disc of radius 2^{-n} around X is equal to the open disc of radius 2^{-n+1} around X and both are equal to the set of all sets Y such that $X|n \subseteq Y$. Therefore, the topology we defined is contained in the metric topology.

For the other containment, let X be any set of consider the set A of all Y such that $d(X, Y) < r$ for some real number r. Let n be the largest n such that $2^{-n} \leq r$. Then A is exactly the set of all Y such that $X|n \subset Y$. Therefore, the metric topology is contained in the topology we defined earlier. □

Lemma 10.2. *For every closed set $C \subset 2^\omega$, there is a binary branching tree T such that $C = [T]$.*

Proof. Recall that the basis of clopen sets for the topology on 2^ω is given by $\mathcal{O}_\sigma = \{X | \sigma \subset X\}$. We define the tree T by

$$\sigma \in T \Leftrightarrow \exists X \in C(\sigma \subset X).$$

That is, $\sigma \in T$ if and only if the basic clopen set $\mathcal{O}_\sigma$ has nontrivial intersection with C.

It remains to show that $C = [T]$. To see the $\supseteq$ containment, let $X \in [T]$. Then, for every n, $X|n \in T$, so we can fix a point $X_n \in C$ such that $X_n \in \mathcal{O}_{X|n}$. Since both X and X_n are in $\mathcal{O}_{X|n}$, $d(X, X_n) \leq 2^{-n}$. Therefore, the points X_n converge to X in the metric topology. Since C is closed and the points X_n are all in C, we must have $X \in C$ as required.

To see the $\subseteq$ containment, let $Y \in C$. Then, for every n, $Y|n \in T$ by the definition of T. Therefore, $Y \in [T]$ as required. □

Lemma 10.2 tells us that we can view all closed sets in 2^ω as sets of paths through binary branching trees. Since Π^0_1 classes are defined as the set of paths through a computable binary branching tree, we can view them as expressing the class of effectively closed sets.

We next show that every computable functional defines a continuous map on 2^ω.

Lemma 10.3. *If Ψ is a computable functional and C is a closed set in 2^ω, then $\Psi^{-1}(C)$ is also a closed set.*

Proof. Fix a tree T such that $C = [T]$. Fix an index e such that $\Psi^Y = \varphi_e^Y$. Since Ψ is a computable functional, $\varphi_e^Y(x)$ converges for all x and all Y. Therefore, for each σ, there is a unique natural number n_σ such that n_σ is the greatest number $\leq |\sigma|$ such that φ_e^σ converges on $0, \ldots, n_\sigma - 1$. We define a tree S by

$$\sigma \in S \Leftrightarrow \varphi_e^\sigma | n_\sigma \in T.$$

S is closed downwards since $\tau \subset \sigma$ implies $n_\tau \leq n_\sigma$ and $\varphi_e^\tau | n_\tau \subset \varphi_e^\sigma | n_\sigma$. Therefore, S is a tree.

It remains to verify that $[S] = \Psi^{-1}([T])$. Notice that

$$Y \in \Psi^{-1}([T]) \Leftrightarrow \varphi_e^Y \in [T] \Leftrightarrow \forall m(\varphi_e^Y | m \in T)$$

and that

$$Y \in [S] \Leftrightarrow \forall m(Y|m \in S) \Leftrightarrow \forall m(\varphi_e^{Y|m} | n_{Y|m} \in T).$$

However, by definition, $\varphi_e^{Y|m} | n_{Y|m} = \varphi_e^Y | n_{Y|m} \in T$. Furthermore, since φ_e^X is total for all X, as $m \to \infty$, $n_{Y|m} \to \infty$. Therefore, $\forall m(\varphi_e^Y | m \in T)$ if and only if $\forall m(\varphi_e^{Y|m} | n_{Y|m})$. □

Having seen that computable functionals define continuous maps, we can ask whether they also define continuous functions if we restrict our attention to effectively closed sets. We prove that Π^0_1 classes of sets are closed under taking inverse images of computable functionals.

Lemma 10.4. *Let $P = [T]$ be a Π^0_1 class of sets and let Ψ be a computable functional. Then $\Psi^{-1}(P)$ is a Π^0_1 class of sets.*

Proof. Fix an index e such that $\Psi^Y = \varphi_e^Y$. By definition, we have

$$Y \in \Psi^{-1}(P) \Leftrightarrow \varphi_e^Y \in P \Leftrightarrow \forall n(\varphi_e^Y|n \in T).$$

Since T is a computable tree and φ_e^Y is total for all Y, the predicate $R(n,Y)$ defined by $\varphi_e^Y|n \in T$ is computable. Therefore, by Proposition 9.5, $Y \in \Psi^{-1}$ defined a Π^0_1 class of sets. □

Projection is another common geometric operation on topological spaces. We next show that Π^0_1 classes are closed under projections by Π^0_1 relations. We use this fact to show that they are also closed under taking images by computable functions.

Lemma 10.5. *Let $R(k,X,Y)$ be a Π^0_1 predicate. The predicate $S(k,X) = \exists Y(R(k,X,Y)$ defines a Π^0_1 class of sets.*

Proof. By the Normal Form Theorem for Π^0_1 predicates, we can write $R(k,X,Y)$ as $\forall n R^*(k,X|n,Y|n)$ where $R^*(k,\sigma,\tau)$ is a computable (even primitive recursive) relation. Therefore, $S(k,X)$ holds if and only if $\exists Y \forall n R^*(k,X|n,Y|n)$. Applying König's Lemma, this statement is equivalent to $\forall n \exists \tau \forall m \leq n(|\tau| = n \wedge R^*(k,X|m,\tau)$. Because $\exists \tau(|\tau| = n)$ is a bounded quantifier, this statement defines a Π^0_1 class of sets. □

Lemma 10.6. *Let P be a Π^0_1 class of sets and let Ψ be a partial computable functional defined at least on all members of P. Then, $\Psi(P)$ is a Π^0_1 class of sets.*

Proof. Fix an index e for Ψ. $X \in \Psi(P)$ if and only if $\exists Y(Y \in P \wedge \forall n(\varphi_e^Y(n) = X(n))$. By definition, $Y \in P$ is Π^0_1. Since $\varphi_e^Y(n)$ is total for all $Y \in P$, we have that $\varphi_e^Y(n) = X(n)$ is a computable check for $Y \in P$. Therefore, by Lemma 10.5, $\Psi(P)$ is a Π^0_1 class. □

In order to establish some measure theoretic results later, we will use yet another characterization of the topology on 2^ω.

Lemma 10.7. *The topology we have defined on 2^ω is equivalent to the product topology on $\{0,1\}^\omega$ where the set $\{0,1\}$ is given the discrete topology.*

Proof. In the product topology, the basic open sets are given by specifying fixed nonempty open sets in a finite number of of coordinates and letting the other coordinates range over $\{0,1\}$. Without loss of generality, we can assume that for any coordinate for which we have fixed an open set, we have fixed either $\{0\}$ or $\{1\}$. Thus, we have just specified a finite number of the coordinates for the elements X in the basic open set. Thus, for every string σ, the set of paths through σ is a basic open set in the product topology. So, our topology is contained in the product topology.

To see the other containment, suppose we have specified finite sets U and V such that our basic open set in the product topology consists of all X such that $X(n) = 1$ for $n \in U$ and $X(n) = 0$ for $n \in V$. Let m be the maximum of $U \cup V$ and let $\sigma_0, \ldots, \sigma_k$ be the set of all strings of length $m+1$ such that $\sigma_i(n) = 1$ if $n \in U$ and $\sigma_i(n) = 0$ if $n \in V$. Then, the basic open set from the product topology is equal to the finite union of $[2^{<\omega}(\sigma_i)]$. (That is, the set of all sets X such that $\sigma_i \subset X$ for some $i \leq k$.) Thus, the product topology is contained in our topology. □

From this description of the topology on 2^ω, it is natural to assign the product measure of the "fair coin flip" measure on $\{0,1\}$ to 2^ω. That is, we assign measure 1/2 to each of the subsets $\{0\}$ and $\{1\}$ in $\{0,1\}$ and then use the product measure on $\{0,1\}^\omega$. Under this measure, the basic open set $\mathcal{O}_\sigma$ has measure 2^{-n} where $n = |\sigma|$. We denote this measure by μ.

We conclude this section with the construction of a Π^0_1 class of sets M that is complete in a sense made specific below. Recall that we have an indexing system for Π^0_1 classes using primitive recursive trees, $P_e = [T_e]$. M is complete in the sense that for any $X \in [M]$ and any e such that $P_e \neq \emptyset$, $X_e \in P_e$, where X_e denotes the e^{th} column of X.

Definition 10.8. Let P and Q be nonempty Π^0_1 classes of sets. We P is **Medvedev reducible** to Q if there is a computable functional Ψ that maps elements of Q to elements of P. We say that Q is **Medvedev complete** if P is Medvedev reducible to Q for all nonempty Π^0_1 classes of sets P.

Lemma 10.9. *There exists a nonempty Π^0_1 class of sets M which is Medvedev complete.*

Proof. Since the list of primitive recursive trees T_e is uniform, there is a computable predicate $U(e,\sigma)$ which holds if and only if $\sigma \in T_e$. Therefore, $X \in [T_e]$ if and only if $\forall n(U(e, X|n))$. We abuse notation slightly be writing $U(e,X)$ in place of $\forall n(U(n, X|n))$. Define the predicate $U^*(e,X)$ by

$$\forall n \forall \sigma \text{ with } |\sigma| = n \left([\forall m \leq n(U(e,\sigma|m))] \rightarrow U(e, X|n) \right).$$

Fix any e such that P_e is nonempty. We claim that $U^*(e,X) \Leftrightarrow U(e,X)$. To see the ($\Leftarrow$) direction, fix X such that $U(e,X)$ holds. For any n, $U(e,X|n)$ holds, so $U^*(e,X)$ holds. To see the ($\Rightarrow$) direction, fix X such that $U(e,X)$ does not hold. $X \notin P_e$ implies that there is an n such that $X|n \notin T_e$. However, since P_e is not empty, there is a $\sigma \in T_e$ with $|\sigma| = n$. This n and σ are witnesses that $U^*(e,X)$ does not hold.

Consider any e such that P_e is empty. If $T_e = \emptyset$, the $U^*(e,X)$ holds for every X. Otherwise, let $\sigma \in T_e$ be such that $\sigma * 0 \notin T_e$ and $\sigma * 1 \notin T_e$. Then, $U^*(e,X)$ holds for any X which extends σ. Therefore, $U^*(e,X)$ is nonempty for every e.

We define M by

$$M = \{\, Y \mid \forall e U^*(e, Y_e) \,\}.$$

(Here, we are again using Y_e to denote the e^{th} column of Y.) M is nonempty because for each e, $U^*(e,X)$ is nonempty. M is a Π^0_1 class because the defining predicate is Π^0_1. M is Medvedev complete because for each nonempty P_e, the computable functional that sends Y to Y_e computes an element in P_e from M. □

11 Degrees of members of Π^0_1 classes

In this section, we examine the Turing degrees of members of Π^0_1 classes of sets. We start with positive results which say that every Π^0_1 class of sets has a member with some degree-theoretic property. (Such results are frequently referred to as "Basis Theorems".)

Definition 11.1. A node σ in a tree T is called **extendible** if there is a path $f \in [T]$ which extends σ. The set of extendible nodes of T is denoted $\mathrm{Ext}(T)$.

Recall König's Theorem for infinite finitely branching trees.

Theorem 11.2. *Every infinite finitely branching tree has a path.*

We are now ready for our first Basis Theorem.

Theorem 11.3. *Every nonempty Π^0_1 class of sets contains a member which is computable from $\mathbf{0}'$.*

Proof. Let $P = [T]$ be a nonempty Π^0_1 class of sets. By König's Theorem, the extendible nodes of T are given by

$$\mathrm{Ext}(T) = \{\, \sigma \in T \mid \forall n \geq |\sigma| \exists \tau (|\tau| = n \wedge \sigma \subset \tau \wedge \tau \in T) \,\}.$$

The quantifier $\exists \tau (|\tau| = n)$ is a bounded quantifier, so the definition of the extendible nodes is Π^0_1 and therefore, the extendible nodes are computable from $\mathbf{0}'$. We can define the "leftmost path" in $[T]$ as follows. Set $f(0) = 0$ if $\langle 0 \rangle$ is extendible and $f(0) = 1$ otherwise. Notice that $\langle f(0) \rangle$ is extendible in either case since $[T]$ is nonempty. Assume that $f(0), \ldots, f(n)$ are defined such that $\langle f(0), \cdots, f(n) \rangle$ is extendible. Define $f(n+1) = 0$ if $\langle f(0), \cdots, f(n), 0 \rangle$ is extendible and $f(n+1) = 1$ otherwise. Notice that the induction hypothesis is satisfied. Furthermore, since the extendible nodes are computable from $\mathbf{0}'$, the leftmost path satisfies $f \leq_T \mathbf{0}'$. □

This proof actually shows something more, which we state as our second Basis Theorem.

Theorem 11.4. *Every nonempty Π^0_1 class of sets contains a member of c.e. degree.*

Proof. Fix a nonempty Π^0_1 class of sets $[T]$ and let L be the leftmost path as defined in the proof of Theorem 11.3. Let N be the set of nodes which lie either on L or to the left of some initial segment of L. We claim that N is an c.e. set. To see the basic idea of why this is so, notice that we can start by enumerating $\langle 0 \rangle$ in L (assuming that it is in T). Only if we later discover that the tree above this node is finite will we we enumerate $\langle 1 \rangle$ into L. Enumerating this node at a late stage into L is not a problem since our only claim is that L is enumerable. The precise details are left to the reader.

Once we know that N is c.e., it only remains to show that $L \equiv_T N$. Clearly, N is computable from L. For the other direction, $\sigma \subset L$ if and only if σ is the rightmost node of length $|\sigma|$ which is in N. □

The next two Basis Theorems use a construction which relies on Examples 9.8 and 9.12. This method of proof is called "forcing with Π^0_1 classes". The first of these theorem is commonly referred to as the Low Basis Theorem.

Theorem 11.5. *Every nonempty Π^0_1 class of sets has a member of low degree.*

Proof. Let $P = [T]$ be a nonempty Π^0_1 class of sets and let Q_e be defined as in Example 9.8. We define a descending sequence of Π^0_1 classes of sets $[S_i]$ contained in P. The sequence of computable binary branching trees S_i will be uniform in $\mathbf{0}'$. Set $S_0 = T$. Assume that S_e has been defined. Ask $\mathbf{0}'$ whether $S_e \cap Q_e$ is finite. That is, ask if

$$\exists n \forall \sigma (|\sigma| = n \rightarrow \sigma \notin S_e \cap Q_e).$$

Because the quantifier $\forall \sigma(|\sigma| = n)$ is a bounded quantifier, this question is Σ^0_1 and can be answered by $\mathbf{0}'$. If the answer is yes, then set $S_{e+1} = S_e$. If the answer is no, then set $S_{e+1} = S_e \cap Q_e$.

Notice that we have forced whether or not $e \in X'$ for all X in $[S_{e+1}]$. That is, $e \in X'$ if and only if $\varphi^X_e(e)\downarrow$. By the definition of Q_e, this is equivalent to $X \notin [Q_e]$. Therefore, if we defined $S_{e+1} = S_e \cap Q_e$, then $X \in [S_{e+1}]$ implies $X \in [Q_e]$ and therefore, $e \notin X'$. However, if we defined $S_{e+1} = S_e$, then $S_e \cap Q_e$ is finite, so $[S_e] \cap [Q_e]$ is empty. Therefore, $X \in [S_{e+1}]$ implies $X \notin [Q_e]$ and hence $e \in X'$.

It is clear that $S_{e+1} \subset S_e$ for all e and hence the sequence of Π^0_1 classes of sets $[S_e]$ is decreasing and contained in P. Furthermore, each $[S_e]$ is a closed subset of 2^ω and therefore, $\cap_{e\in\omega}[S_e]$ is nonempty. (Notice that although each $[S_e]$ is a Π^0_1 class of sets, the intersection is not a Π^0_1 class of sets because the sequence S_e is not computable; it is only computable in $\mathbf{0}'$.) It remains to show that every member of this intersection has low degree. At the e^{th} stage of the construction, we forced whether or not $e \in X'$. Since the construction is computable in $\mathbf{0}'$, we can determine if $e \in X'$ computablely in $\mathbf{0}'$. □

For the next Basis Theorem, we need to recall a definition from classical recursion theory.

Definition 11.6. A set A is **hyperimmune-free** if for every total function $f \leq_T A$, there is a computable function g such that $f(n) \leq g(n)$ for all n. In other words, every total function computable in A is majorized by a computable function.

Theorem 11.7. *Every nonempty Π^0_1 class of sets contains a member of hyperimmune-free degree.*

Proof. Let $P = [T]$ be a nonempty Π^0_1 class of sets. We need to find a set $A \in P$ such that for all e, if φ^A_e is total, then φ^A_e is majorized by a computable function. We force with a slightly different set of Π^0_1 classes. For each e and i, let Q^i_e be as in Example 9.8. Recall that $[Q^i_e]$ is the set of all X for which $\varphi^X_e(i)\uparrow$.

As in the proof of the Low Basis Theorem, we define a decreasing sequence of computable binary branching trees S_e contained in T. As above, the intersection $\cap[S_e]$ will be nonempty and each set A in this intersection will be hyperimmune-free. For this proof, we do not need to worry about the complexity of defining the decreasing sequence of Π^0_1 classes.

Set $S_0 = T$. Assume that S_e has been defined. Ask if $S_e \cap Q^i_e$ is finite for all i. If so, then set $S_{e+1} = S_e$. If not, then pick an i such that $S_e \cap Q^i_e$ is infinite and set $S_{e+1} = S_e \cap Q^i_e$.

Notice that we have again forced our desired result for index e at stage $e+1$. If $S_{e+1} = S_e \cap Q^i_e$, then φ^A_e is not total for any $A \in [S_{e+1}]$ since $\varphi^A_e(i)\uparrow$. If $S_{e+1} = S_e$, then we know φ^A_e is total for all $A \in [S_{e+1}]$. To define the majorizing function $f(x)$, search for a level n such that $\varphi^\sigma_e(x)\downarrow$ for all $\sigma \in S_{e+1}$ with $|\sigma| = n$. (If there were no such level, then every level would contain a string σ for which $\varphi_e^\sigma(x)\uparrow$. But, then the set of such nodes would form an infinite Π^0_1 subtree of $S_e = S_{e+1}$ which would be contained in Q_e^i, contrary to the definition of S_{e+1}.) Set $f(x)$ equal to the maximum of $\varphi_e^\sigma(x)$ for $\sigma \in S_{e+1}$ with $|\sigma| = n$. The function f is the desired majorizing function. □

Theorem 11.8. *Let P be a nonempty Π^0_1 class of sets and let C be any noncomputable set. There is an $X \in P$ for which $C \not\leq_T X$.*

Proof. Let $P = [S_0]$ and we define a decreasing sequence of infinite trees $S_{e+1} \subset S_e$ as in the last two theorems. We want to make sure that for all $X \in [S_{e+1}]$, $\varphi_e^X \neq C$. Assume that we have defined S_e. There are several cases to consider for the definition of S_{e+1}. (Once again, we do not need to worry about the complexity of the questions we ask when defining S_{e+1}, since in each case we define S_{e+1} from S_e plus a finite amount of information.)

First, we ask if

$$\forall n \exists i \forall X \in [S_e](\varphi_e^X(n) = i).$$

If the answer is yes, then for all paths $X \in [S_e]$, φ_e^X is the same function. Furthermore, this function is computable. To compute $\varphi_e^X(n)$, look for a level in S_e such that for every $\sigma \in S_e$ at this level, $\varphi_e^\sigma(n)$ converges and gives the same answer. We know such a level exists, since if not, there would be infinitely many $\sigma \in S_e$ for which $\varphi_e^\sigma(n)$ either does not converge or converges to a different value. By König's Lemma, there would be a path X which contradicts the answer to our question above. Since C is not computable and every $X \in [S_e]$ satisfies φ_e^X is computable, we can set $S_{e+1} = S_e$.

Second, assume that the answer to the question above is no. We next ask if there is an n such that $Q_e^n \cap S_e$ is infinite. If so, then we set $S_{e+1} = Q_e^n \cap S_e$. As in the last theorem, we have forced that every $X \in [S_{e+1}]$ satisfies φ_e^X is not total (and therefore cannot equal C).

Third, if the answer to this second question is also no, then as in the last theorem, we know that for all $X \in [S_e]$, φ_e^X is total. However, since the answer to our first question was no, we know that there are $X_1, X_2 \in [S_e]$ and $n \in \omega$ such that $\varphi_e^{X_1}(n) \neq \varphi_e^{X_2}(n)$. One of these values must differ from $C(n)$. Without loss of generality, assume that $\varphi_e^{X_1}(n) \neq C(n)$. Let σ be an initial segment of X_1 such that $\varphi_e^\sigma(n)$ converges. Let S_{e+1} be the set of all nodes in S_e which are comparable to σ. We know $[S_{e+1}]$ is not empty since $X_1 \in [S_{e+1}]$ and we know for all $Y \in [S_{e+1}]$, $\varphi_e^Y(n) = \varphi_e^{X_1}(n) \neq C(n)$. So, again we are done.

As in the previous two theorems, $\cap S_e$ is nonempty. For each X in this intersection, $\varphi_e^X \neq C$ because of our forcing when we define S_{e+1}. □

Consider for a moment the special case of a countable Π^0_1 class $P = [T]$ of sets. If each $\sigma \in \mathrm{Ext}(T)$ had two incompatible extension in $\mathrm{Ext}(T)$, then T would have $2^{\aleph_0}$ many paths. Therefore, there must be a node $\sigma \in \mathrm{Ext}(T)$ such that there is a unique node $\tau \in \mathrm{Ext}(T)$ at each level above σ. That is, there is a unique path f in T passing through σ. Or, in other words, there is an isolated path in T. Since any isolated path is computable, we have shown the following lemma.

Lemma 11.9. *Every countable Π^0_1 class of sets contains a computable member.*

It is not the case that every Π^0_1 class of sets contains a computable member.

Theorem 11.10. *There is a nonempty Π^0_1 class P of sets which contains no computable member.*

Proof. Let P be the Π^0_1 class of DNR_2 functions from Example 9.10. By the diagonal nature of its definition, this class cannot contain a computable set. For this theorem, it would suffice to take $\mathrm{Sep}(A, B)$ for any pair of computablely inseparable c.e. sets A and B. □

If $[T]$ contains no isolated paths, then $[T]$ is called **perfect**. In this case, any $\sigma \in \mathrm{Ext}(T)$ has incomparable extensions $\tau_0, \tau_1 \in \mathrm{Ext}(T)$, and therefore $[T]$ has size $2^{\aleph_0}$. Combining this observation with Lemma 11.9, we see that any Π^0_1 class of sets that does not have a computable member must have size $2^{\aleph_0}$. We can say much more about the degrees that occur in such a class. Before we get to this statement, we prove a lemma which is reminiscent of the computation lemmas in the construction of a minimal Turing degree.

Definition 11.11. Let T be a binary branching tree. A node $\sigma \in \mathrm{Ext}(T)$ is said to **e-split** if there are extensions τ_1 and τ_2 of σ in $\mathrm{Ext}(T)$ such that $\varphi_e^{\tau_1}(x) \downarrow \neq \varphi_e^{\tau_2}(x) \downarrow$ for some x. The tree T is said to be **e-splitting** if every $\sigma \in \mathrm{Ext}(T)$ e-splits.

If T is a binary branching tree and $\sigma \in T$, then we let $T(\sigma)$ denote the subtree of T which consists of the nodes in T that are comparable with σ. That is, $T(\sigma)$ consists of the initial segment σ followed by all nodes in T that sit above σ.

Lemma 11.12. *Let T be an infinite computable binary branching tree and let e be any index. One of the following three situations must occur:*

1. *T is e-splitting; or*
2. *there is an x for which $Q_e^x \cap T$ is infinite; (recall the definition of Q_e^x from Example 9.8) or*
3. *there is a $\sigma \in \mathrm{Ext}(T)$ such that for every $f \in [T(\sigma)]$, φ_e^f is total and computable.*

Proof. Assume that both (1) and (2) fail. Since (1) fails, we can fix a $\sigma \in \mathrm{Ext}(T)$ such that for any extensions $\tau_1, \tau_2 \in \mathrm{Ext}(T)$ of σ and any x, if both $\varphi_e^{\tau_1}(x)$ and $\varphi_e^{\tau_2}(x)$ converge, then they are equal. Since (2) fails, for each x there are only finitely many $\tau \in T$ for which $\varphi_e^{\tau}(x)$ does not converge. We show there is a computable function g such that for all $f \in [T(\sigma)]$, φ_e^f is total and $\varphi_e^f = g$.

Fix any value x and we define $g(x)$. Since there are only finitely many $\tau \in T$ for which $\varphi_e^{\tau}(x)$ fails to converge, there is a level m such that for all $\tau \in T$ with $|\tau| = m$, $\varphi_e^{\tau}(x)$ converges. By searching with dovetailed computations, we can find the value of m. Furthermore, any pair of extendible nodes at this level must agree on the value of their computation. Therefore, we can search for a level $n \geq m$ at which all nodes $\tau \in T$ with $|\tau| = n$ agree on the value of φ_e^{τ}. Once we find such a level, we set $g(x)$ equal to the value of φ_e^{τ} for any $\tau \in T$ with $|\tau| = n$. □

Lemma 11.13. *If T is an infinite computable binary branching tree such that $[T]$ has no computable member, then for every infinite computable subtree $S \subset T$, $[S]$ is perfect and therefore $|[S]| = 2^{\aleph_0}$.*

Proof. If $[S]$ is not perfect, then $[S]$ has an isolated member which is computable, which contradicts the fact that $[T]$ has no computable member. $\square$

Definition 11.14. A **rooted tree** is a pair $\langle \sigma, T\rangle$ such that $T = T(\sigma)$. That is, T restricted to level $|\sigma|$ is the linear segment σ. The rooted tree $\langle \sigma_1, T_1\rangle$ **extends** $\langle \sigma_0, T_0\rangle$ if $\sigma_0 \subset \sigma_1$ and $T_1 \subset T_0$. $\langle \sigma_0, T_0\rangle$ and $\langle \sigma_1, T_1\rangle$ are **incomparable** if σ_0 and σ_1 are incomparable. A binary function f is a path in $\langle \sigma, T\rangle$ if f is a path in T.

We write $\langle \sigma_0, T_0\rangle \subset \langle \sigma_1, T_1\rangle$ to denote that $\langle \sigma_1, T_1\rangle$ extends $\langle \sigma_0, T_0\rangle$.

Theorem 11.15. *Let $P = [T]$ be a Π^0_1 class of sets without a computable member and let $\{\mathbf{a_i} | i \in \omega\}$ be any countable sequence of noncomputable degrees. P has $2^{\aleph_0}$ members f which are mutually Turing incomparable and which are incomparable with each $\mathbf{a_i}$.*

Proof. Fix sets A_i with degree $\mathbf{a_i}$. We define a sequence $\mathcal{R}_e$, $e \in \omega$, such that each $\mathcal{R}_e$ is a collection of 2^e pairwise incomparable infinite computable rooted trees contained in T and such that each rooted tree in $\mathcal{R}_e$ has exactly two extensions in $\mathcal{R}_{e+1}$. We set $\mathcal{R}_0$ to be the singleton set $\{\langle \lambda, T\rangle\}$. The sets $\mathcal{R}_e$ will satisfy the following requirements.

1. If f belongs to one of the rooted trees in $\mathcal{R}_{e+1}$, then for all pairs $\langle i, j\rangle \leq e$, $\varphi^f_i \neq A_j$.

2. If f and g belong to different rooted trees of $\mathcal{R}_{e+1}$, then for every $i \leq e$, $\varphi^f_i \neq g$.

Notice that since each rooted tree in $\mathcal{R}_n$ is an infinite computable subtree of T, each such tree has no isolated paths by Lemma 11.13. Therefore, for any σ which is an extendible node in one of there trees, there are incomparable extensions of σ which are themselves extendible.

First we show that these requirements are enough to prove the theorem. Let $\mathcal{C}$ be the set of all paths f in $[T]$ such that for every n, f is in some rooted tree in $\mathcal{R}_n$. Notice that $\mathcal{C}$ has size $2^{\aleph_0}$ since each rooted tree in $\mathcal{R}_n$ has two incomparable extensions in $\mathcal{R}_{n+1}$. By requirement (1), no element of $\mathcal{C}$ can compute any A_i. By requirement (2), if $f \neq g \in \mathcal{C}$, then $\varphi^f_e \neq g$ for any e. Finally, let $\mathcal{D} \subset \mathcal{C}$ be the collection of functions in $\mathcal{C}$ which are not computable from any set A_i. Since $\mathcal{C}$ and $\mathcal{D}$ differ by a countable set, $|\mathcal{D}| = 2^{\aleph_0}$. Therefore, $\mathcal{D}$ is the required set of paths in $[T]$.

It remains to show how to satisfy the requirements. We define $\mathcal{R}_0 = \{\langle \lambda, T\rangle\}$. Assume that $\mathcal{R}_e$ has been defined. For each rooted tree $\langle \sigma, R\rangle$ in $\mathcal{R}_e$, we will define extensions $\langle \tau_i, R_i\rangle$ and $\langle \delta_i, S_i\rangle$ for $0 \leq i \leq 2$ such that

$$\langle \tau_0, R_0\rangle \subset \langle \tau_1, R_1\rangle \subset \langle \tau_2, R_2\rangle$$
$$\langle \delta_0, S_0\rangle \subset \langle \delta_1, S_1\rangle \subset \langle \delta_2, S_2\rangle.$$

The extensions indexed with 0 will be chosen to make sure that these τ_0 and δ_0 are incomparable. The extensions indexed with a 1 will be chosen to satisfy requirement (1). The extensions indexed with 2 will be chosen to satisfy requirement (2) and they will be the extensions that are put into $\mathcal{R}_{e+1}$.

Let $\langle \sigma, R\rangle$ be an arbitrary rooted tree in $\mathcal{R}_e$ and assume that $e = \langle i, j\rangle$. We choose incomparable extensions τ_0 and δ_0 of σ which are extendible in R. We set $R_0 = R(\tau_0)$ and $S_0 = R(\delta_0)$.

To satisfy requirement (1) for τ_0, we apply Lemma 11.12 to R_0 with index i. Our goal is to define an extension $\langle \tau_1, R_1 \rangle$ of $\langle \tau_0, R_0 \rangle$ which will serve as our second approximation to the extension we will put into $\mathcal{R}_{e+1}$.

If condition (1) from Lemma 11.12 holds, then there must be an x and extensions γ_0 and γ_1 of τ_0 such that $\varphi_i^{\gamma_0}(x) \downarrow \neq \varphi_i^{\gamma_1}(x) \downarrow$. One of these computations must disagree with $A_j(x)$. Without loss of generality, assume $\varphi_i^{\gamma_0}(x) \neq A_j(x)$. In this case, we set $\tau_1 = \gamma_0$ and $R_1 = R_0(\gamma_0)$. Notice that for any path f in $\langle \tau_1, R_1 \rangle$, $\varphi_i^f(x) \neq A_j(x)$ as required.

If condition (2) from Lemma 11.12 holds, then fix an x such that $Q_i^x \cap R_0$ is infinite. Set $\tau_1 = \tau_0$ and $R_1 = Q_i^x \cap R_0$. Notice that for any path f in $\langle \tau_1, R_1 \rangle$, $\varphi_i^f(x)$ does not converge, and therefore cannot be equal to $A_j(x)$.

If condition (3) from Lemma 11.12 holds, then set $\tau_1 = \sigma$ and $R_1 = R_0(\sigma)$, where σ is as in condition (3) of Lemma 11.12. For any path f in $\langle \tau_1, R_1 \rangle$, φ_i^f is computable and hence cannot be equal to A_j, which is noncomputable.

We perform the same actions for δ_0 and S_0 to get a second approximation $\langle \delta_1, S_1 \rangle$ to its rooted tree which satisfies requirement (1). Assume that we have defined these second approximations for all the rooted trees in $\mathcal{R}_e$. We need to satisfy requirement (2) between these trees.

To satisfy requirement (2) between the trees $\langle \tau_1, R_1 \rangle$ and $\langle \delta_1, S_1 \rangle$, fix an index $k \leq e$. Apply Lemma 11.12 to R_1 with index k. If case (1) of Lemma 11.12 applies, then fix an x and extensions γ_0 and γ_1 of τ which are extendible such that $\varphi_k^{\gamma_0}(x) \downarrow \neq \varphi_k^{\gamma_1}(x) \downarrow$. If necessary, extend δ_1 to a node in $\delta_1' \in S_1$ which has length $> x$. Without loss of generality, assume that $\varphi_k^{\gamma_0}(x) \neq \delta_1'(x)$. Extend $\langle \tau_1, R_1 \rangle$ to $\langle \gamma_0, R_1(\gamma_0) \rangle$ and $\langle \delta_1, S_1 \rangle$ to $\langle \delta_1', S_1(\delta_1') \rangle$. These extended rooted trees satisfy requirement (2) for the index k.

If condition (2) of Lemma 11.12 applies, then pick an x such that $Q_e^x \cap R_1$ is infinite and extend $\langle \tau_1, R_1 \rangle$ to $\langle \tau_1, Q_e^x \cap R_1 \rangle$. Since $\varphi_k^f(x)$ does not converge for any path in this extended rooted tree, it satisfies requirement (2) with $\langle \delta_1, S_1 \rangle$ for index k.

If condition (3) of Lemma 11.12 applies, then fix σ from that condition. Extend $\langle \tau_1, R_1 \rangle$ to $\langle \sigma, R_1(\sigma) \rangle$. Since φ_k^f is computable for any path in this extended rooted tree and since $\langle \delta_1, S_1 \rangle$ does not contain any computable paths (remember that it is a subtree of the original tree T), we have again satisfied requirement (2) with the index k.

Notice that we are far from through with requirement (2). For each pair of second level approximations and each index $k \leq e$, we need to perform the above extensions. Once $\langle \tau_1, R_1 \rangle$ has been compared to every other second level approximation with each index $k \leq e$, we let $\langle \tau_2, R_2 \rangle$ be the result and we put this rooted tree into $\mathcal{R}_{e+1}$. □

This theorem shows that if P is a Π_1^0 class of sets without a computable member, then the set of degrees which occur among the members of P cannot be easily described. For example, it cannot be a cone of degrees or even a countable union of cones of degrees.

Corollary 11.16. *For any Π_1^0 class of sets P with at least two members and any $X \in P$, there is a $Y \in P$ such that $deg(X) \wedge deg(Y) = \mathbf{0}$.*

Proof. If P has a computable member, then let Y be a computable set in P. Otherwise, apply Theorem 11.15 with the noncomputable degrees $\leq_T \deg(X)$. This produces a Y that is not above any noncomputable degree below X. Therefore, any set computable from both X and Y must be computable. □

12 Applications

We have already seen the following theorems.

Theorem 12.1. *For any c.e. Boolean algebra B there is a computable binary branching tree T_B and a Turing degree preserving bijection between the ultrafilters in B and the members of $[T_B]$.*

Theorem 12.2. *For any consistent computable theory T, there is a computable binary branching tree S_T and a Turing degree preserving bijection between the complete consistent extensions of T and the members in $[S_T]$.*

Most of the results from the last section have the form "Every Π^0_1 class has a member with some specified property". We can combine these results with Theorems 12.1 and 12.2 to conclude facts about ultrafilters in c.e. Boolean algebra and complete consistent extensions of computable theories.

Theorem 12.3. *Let B be an infinite c.e. Boolean algebra.*

1. *B has an ultrafilter computable from $\mathbf{0}'$.*
2. *B has an ultrafilter of c.e. degree.*
3. *B has an ultrafilter of low degree.*
4. *B has an ultrafilter of hyperimmune-free degree.*
5. *For every noncomputable set C, B has an ultrafilter which does not compute C.*
6. *If B has only countably many ultrafilters, then B has a computable ultrafilter.*
7. *If B does not have a computable ultrafilter, then for any computable sequence of degrees $\mathbf{a_i}$, $i \in \omega$, B has $2^{\aleph_0}$ many ultrafilters which are pairwise Turing incomparable and which are all incomparable with each $\mathbf{a}_i$.*
8. *For any ultrafilter U_1 on B, there is an ultrafilter U_2 on B such that $deg(U_1) \wedge deg(U_2) = \mathbf{0}$.*

Proof. These are all direct consequences of the theorems from the previous section. (1) follows from Theorem 11.3, (2) follows from Theorem 11.4, (3) follows from Theorem 11.5, (4) follows from Theorem 11.7, (5) follows from Theorem 11.8, (6) follows from Lemma 11.9, and (7) follows from Theorem 11.15. Because B is an infinite algebra, B has at least two ultrafilters and therefore (8) follows from Corollary 11.16. □

Theorem 12.4. *Let T be a computable consistent theory.*

1. T has a complete consistent extension computable from $\mathbf{0}'$.

2. T has a complete consistent extension of c.e. degree.

3. T has a complete consistent extension of low degree.

4. T has a complete consistent extension of hyperimmune-free degree.

5. For any noncomputable set C, T has a complete consistent extension which does not compute C.

6. If T has only countably many complete consistent extensions, then B has a computable complete consistent extension.

7. If T does not have a computable complete consistent extension, then for any computable sequence of degrees $\mathbf{a_i}$, $i \in \omega$, T has $2^{\aleph_0}$ many complete consistent extensions which are pairwise Turing incomparable and which are all incomparable with each $\mathbf{a}_i$.

8. If T has at least two complete consistent extensions, then for any complete consistent extension T_1 of T, there is a complete consistent extension T_2 of T such that $deg(T_1) \wedge deg(T_2) = \mathbf{0}$.

There are lots of common theories to which this theorem applies. For example, you can apply to both PA and ZFC.

Notice that we did not apply Theorem 11.10 to either c.e. Boolean algebras or consistent computable theories. Theorem 11.10 has the form "There is a Π^0_1 class of sets with some particular property". In order to apply such a result, we need our representation theorems for c.e. Boolean algebras and consistent computable theories to go the other direction. That is, we need to show that for each nonempty Π^0_1 class of sets P, there is an c.e. Boolean algebra B and a consistent computable theory T such that the ultrafilters of B and the complete consistent extensions of T are in Turing degree preserving bijective correspondence with the elements of P.

In general, there are a number of ways to consider representing Π^0_1 classes, and we will start with the strongest possible interpretation of this idea. Assume that we have some combinatorial or algebraic problem, such as finding a complete consistent extension of a consistent computable theory or finding an ultrafilter in an c.e. Boolean algebra. We say that this problem **strongly represents** all Π^0_1 classes of sets if for every Π^0_1 class of sets there is an instance of problem such that the class of solutions to the particular instance of the problem is in a bijective Turing degree preserving correspondence with the members of the Π^0_1 class.

Theorem 12.5. *For every nonempty Π^0_1 class of sets P, there is a consistent computable theory Γ_P such that the set of complete consistent extensions of Γ_P is in a bijective Turing degree preserving correspondence with the members of P. In other words, the sets of complete consistent extensions of computable theories can strongly represent all Π^0_1 classes.*

Proof. Let A_n be a countable list of propositional variables, which we use as our computable propositional language $\mathcal{L}$. We let A_n^1 denote A_n and A_n^0 denote $\neg A_n$. We can define a complete consistent extension C_X from any set X by setting $A_n^{X(n)} \in C_X$ for all n. We can close C_X under propositional logical consequence in a computable way (using truth tables), so this correspondence preserves Turing degree. We define a computable theory Γ_P such that for any set X, C_X is a complete consistent extension of Γ_P if and only if $X \in P$.

Let $P = [T]$ where T is a computable binary branching tree. For any string σ of length n, we let P_σ denote the propositional formula $A_0^{\sigma(0)} \wedge \cdots \wedge A_{n-1}^{\sigma(n-1)}$. We define Γ_P by

$$P_\sigma \rightarrow A_n^{1-i} \Leftrightarrow \sigma \in T \wedge |\sigma| = n \wedge \sigma * i \notin T.$$

To understand what this axiom says, recall that $\sigma \in T$ means that we currently think P_σ should be contained in some complete consistent extension. If $\sigma * i \notin T$, then we think that any extension of P_σ should not be an extension of $P_{\sigma*i}$. Therefore, we want to make $P_{\sigma*i}$ inconsistent (relative to T_P) with any extension that contains P_σ. We can accomplish this by specifying that any extension containing P_σ must contain A_n^{1-i}. This is exactly what the axiom above says.

We need to verify that $X \in [T]$ if and only if C_X is a complete consistent extension of Γ_P. Since C_X is both complete and consistent, we only need to show that it is a model of Γ_P if and only if $X \in [T]$. First, suppose that $X \notin [T]$ and fix n such that $X|n \in T$ but $X|(n+1) \notin T$. Then $P_{X|n} \rightarrow A_n^{1-X(n)}$ is an axiom in Γ_P. Since $P_{X|n}, A_n^{X(n)}$ are in C_X, it is clear that C_X is not a model of Γ_P.

Second, assume that $X \in [T]$. We need to show that C_X is a model of Γ_P. Consider an axiom $P_\sigma \rightarrow A_n^{1-i}$ of Γ_P where $|\sigma| = n$. If $\sigma \neq X|n$, then C_X is not a model for P_σ, and hence C_X satisfies $P_\sigma \rightarrow A_n^{1-i}$. If $\sigma = X|n$, then C_X is a model for P_σ. Because $\sigma * i \notin T$, we know that any path in $[T]$ through σ must pass through $\sigma * (1-i)$. Therefore $X(n) = 1 - i$, which implies C_X is a model for A_n^{1-i} and also for $P_\sigma \rightarrow A_n^{1-i}$. □

There is nothing special about using a propositional theory here. We could also have done the coding by a predicate theory with a single binary relation.

Next we use the Lindenbaum algebras of the theories just constructed to prove that spaces of ultrafilters on c.e. Boolean algebras can represent any Π_1^0 class.

Corollary 12.6. *For any Π_1^0 class of sets P, there is an c.e. Boolean algebra B_P such that the space of ultrafilters on B_P is in a bijective Turing degree preserving correspondence with the members of P. In other words, the space of ultrafilters on c.e. Boolean algebras can strongly represent any Π_1^0 class of sets.*

Proof. Let T_P be the theory from Theorem 12.5 and let B_P be the Lindenbaum algebra of this theory. We have already seen that B_P is an c.e. Boolean algebra such that the ultrafilters of B_P are in bijective Turing degree preserving correspondence with the complete consistent extensions of T_P. Therefore, by Theorem 12.5, the ultrafilters are in bijective Turing degree preserving correspondence with the members of P. □

Unfortunately, we cannot relax the complexity requirement on the Boolean algebras in Corollary 12.6 from c.e. to computable.

Lemma 12.7. *There is a Π^0_1 class of sets P such that there is no computable Boolean algebra B for which the ultrafilters on B are in bijective Turing degree preserving correspondence with the members of P.*

Proof. From Theorem 11.10, we know that there are Π^0_1 classes of sets with no computable members. However, we also know that every computable Boolean algebra has a computable ultrafilter. Therefore, no computable Boolean algebra can strongly represent the members of a Π^0_1 class of sets with no computable members. □

13 Exotic Π^0_1 classes

In this section, we give more examples of theorems like Theorem 11.10 which state the existence of Π^0_1 classes with particular properties. In each case, we state the corollaries which follows immediately from Theorem 12.5 and Corollary 12.6.

Theorem 13.1. *There is a nonempty Π^0_1 class of sets P such that for any $X \in P$ and any c.e. set C, if $X \leq_T C$, then $C \equiv_T \mathbf{0}'$.*

Proof. Let P be the class of DNR_2 sets from Example 9.10. Fix any $X \in P$ and any c.e. set C such that $X \leq_T C$. The Arslanov Completeness Criterion says that $C \equiv_T \mathbf{0}'$ if there is a function $f \leq_T C$ for which $W_e \neq W_{f(e)}$ for all e. (W_e is the standard enumeration of the c.e. sets by $W_e = \text{dom}(\varphi_e)$.) Therefore, since $X \leq_T C$, it suffices to show that we can construct such an f computable in X.

First, we define a partial computable function $g(e, x)$ such that $g(e, x) = 1$ if there is an s such that $\varphi_{e,s}(1)$ converges and $\varphi_{e,s}(0)$ does not converge; $g(e, x) = 0$ if there is an s such that $\varphi_{e,s}(0)$ converges and $\varphi_{e,s}(1)$ does not; and $g(e, x)$ is undefined otherwise. Informally, $g(e, x)$ runs the computations $\varphi_e(0)$ and $\varphi_e(1)$ simultaneously and outputs 0 or 1 depending on which converges first. If neither converges, then $g(e, x)$ does not halt. Notice that $g(e, x)$ only depends on e and is constant with respect to x.

By the *s-m-n*-Theorem, there is a total computable function $h(e)$ such that $\varphi_{h(e)}(x) = g(e, x)$ for all e and x. Therefore, if $\varphi_e(0)$ halts first, then $\varphi_{h(e)}(x) = 0$ for all x; if $\varphi_e(1)$ halts first, then $\varphi_{h(e)}(x) = 1$ for all x; and if neither computation halts, then $\varphi_{h(e)}$ is the constantly undefined function.

Notice that if $1 \in W_e$ and $0 \notin W_e$, then $\varphi_{h(e)}(h(e)) = 1$. Because X is a DNR_2 set, this implies that $h(e) \notin X$. Similarly, if $0 \in W_e$ and $1 \notin W_e$, then $\varphi_{h(e)}(h(e)) = 0$ and so $h(e) \in X$.

We can now define our function f from X. If $h(e) \in X$, then we let $f(e)$ be an index for an c.e. set such that $W_{f(e)} = \{1\}$. If $h(e) \notin X$, then we let $f(e)$ be an index for an c.e. set such that $W_{f(e)} = \{0\}$.

To check that these definitions give $W_e \neq W_{f(e)}$ for all e, notice that $W_{f(e)}$ is always either the singleton $\{0\}$ or the singleton $\{1\}$. Therefore, unless W_e is one of these sets, we have succeeded trivially. Suppose $W_e = \{0\}$. Then $0 \in W_e$ and $1 \notin W_e$, so $h(e) \in X$ and $W_{f(e)} = \{1\} \neq W_e$. Similarly, if $W_e = \{1\}$, then $1 \in W_e$ and $0 \notin W_e$, so $h(e) \notin X$ and $W_{f(e)} = \{0\} \neq W_e$. □

Corollary 13.2. *There is an infinite c.e. Boolean algebra B and a complete consistent theory T such that for every ultrafilter U on B of c.e. degree and every complete consistent extension S of T of c.e. degree satisfies $U \equiv_T S \equiv_T \mathbf{0}'$.*

We also get the following interesting, purely computability theoretic, corallary. This corollary follows from the fact that the set X in the proof of Theorem 13.1 can be chosen to be low.

Corollary 13.3. *There is a low degree $\mathbf{d}$ such that the only c.e. degree above $\mathbf{d}$ is $\mathbf{0}'$.*

Theorem 13.4. *There is an infinite Π^0_1 class of sets P such that for all $X \neq Y \in P$, X and Y are Turing incomparable.*

Proof. We define P via a computable sequence of computable functions ψ_s from $2^{<\omega}$ into $2^{<\omega}$. For each ψ_s we require that

1. $\psi_s(\sigma * 0)$ and $\psi_s(\sigma * 1)$ are incompatible extensions of $\psi_s(\sigma)$ for all σ,
2. $\text{range}(\psi_{s+1}) \subset \text{range}(\psi_s)$, and
3. $\lim_s \psi_s(x) = \psi(x)$ exists for all x.

By (1), we can define a computable tree T_s as the downward closure of range(ψ_s). That is, $\tau \in T_s$ if and only if $\exists \sigma(\tau \subset \psi_s(\sigma))$. Since ψ_s is total, $|[T_s]| = 2^{\aleph_0}$. By (2), we have that $T_{s+1} \subset T_s$, so we can define a nonempty Π^0_1 class of sets $P = [T]$ where $T = \cap T_s$. By (3), $|[T]| = 2^{\aleph_0}$.

To guarantee that any pair of distinct sets in $[T]$ are Turing incomparable, we meet the requirements

$$R_e : \forall X \in [T]\Big((\varphi_e^X \text{ total} \wedge \varphi_e^X \neq X) \rightarrow \varphi_e^X \notin [T]\Big).$$

We split R_e into subrequirments labeled by binary strings σ of length $e+1$. For each such σ, R_e^σ is the requirement

$$\forall X \in [T(\psi(\sigma))]\Big((\varphi_e^X \text{ total} \wedge \varphi_e^X \neq X) \rightarrow \varphi_e^X \notin [T]\Big).$$

The requirement R_e^σ may change the values of $\psi_s(\tau)$ for $|\tau| > e$, but it will never change the values for strings of length $\leq e$. Therefore, our action for R_e cannot injure any R_i with $i < e$. The construction will be finite injury, which will guarantee property (3).

If $\varphi_e^\sigma(y)$ converges for all $y \leq x$, then we say $\varphi_e^\sigma(0, \ldots, x)$ is defined and we let $\varphi_e^\sigma(0, \ldots, x)$ denote the string $\langle \varphi_e^\sigma(0), \ldots, \varphi_e^\sigma(x)\rangle$. We say that R_e^σ is satisfied at stage s if there is an x such that $\varphi_{e,s}^{\psi_s(\sigma)}(0, \ldots, x)$ is defined and $\notin T_s$. If R_e^σ is satisfied at stage s and $\psi_s(\sigma) = \psi(\sigma)$, then R_e^σ is satisfied at every stage $t > s$ and hence in $[T]$. Therefore, once the requirements R_i^τ with $i < e$ have stopped acting, R_e^σ will act at most once.

We say that R_e^σ requires attention at stage $s+1$ if R_e^σ is not satisfied at stage $s+1$ and there exists a σ' extending σ (with $|\sigma'| \leq s$) and an $x \leq s$ such that $\varphi_{e,s}^{\psi_s(\sigma')}(0, \ldots, x)$ is defined and incompatible with $\psi_s(\sigma')$, and $\varphi_e^{\psi(\sigma')}$ extends $\psi_s(\rho * i)$ for some $|\rho| = e+1$ and $i \in \{0, 1\}$.

We can now describe the construction. At stage 0, set $\psi_0(\sigma) = \sigma$ for all σ. At stage $s+1$, check if any requirement R_e^σ with $e \leq s$ requires attention. If there are no such requirements, let $\psi_{s+1} = \psi_s$. Otherwise, let R_e^σ be the highest priority requirement that requires attention. (We order the requirements first by e and then by some fixed computable listing of the strings σ.)

Assume R_e^σ requires attention and σ', ρ and i are as above. If $\rho \neq \sigma$, then let $\psi_{s+1}(\mu) = \psi_s(\sigma' * \mu')$ if $\mu = \sigma * \mu'$, $\psi_{s+1}(\mu) = \psi_s(\rho * (1-i) * \mu')$ if $\mu = \rho * \mu'$, and $\psi_{s+1}(\mu) = \psi_s(\mu)$ otherwise. Notice that, as claimed above, for all μ with $|\mu| \leq e$, $\psi_{s+1}(\mu) = \psi_s(\mu)$.

Consider what the action of the last paragraph does. We know that $\varphi_e^{\psi_s(\sigma')}(0,\dots,x)$ extends $\psi_s(\rho * i)$. However, $\psi_s(\rho * i)$ is incompatible with $\psi_{s+1}(\rho) = \psi_s(\rho * (1-i))$, and therefore, $\varphi_e^{\psi_s(\sigma')}(0,\dots,x) \notin T_{s+1}$. R_e^σ is thus satisfied at stage $s+1$ after this action.

If $\rho = \sigma$, then define $\psi_{s+1}(\mu) = \psi_s(\sigma' * \mu')$ if $\mu = \sigma * \mu'$ and $\psi_{s+1}(\mu) = \psi_s(\mu)$ otherwise. Again, as claimed above, for all μ with $|\mu| \leq e$, $\psi_{s+1}(\mu) = \psi_s(\mu)$. This ends the construction at stage $s+1$.

Consider what this last action accomplishes. We know $\varphi_e^{\psi_s(\sigma')}(0,\dots,x)$ is incompatible with $\psi_s(\sigma')$ and $\varphi_e^{\psi_s(\sigma')}(0,\dots,x)$ extends $\psi_s(\sigma)$. Therefore, there is an $n \leq x$ such that $|\psi_s(\sigma)| \leq n < |\psi_s(\sigma')|$ and $\varphi_e^{\psi_s(\sigma')}(n) \neq \psi_s(\sigma')$. However, in T_{s+1}, there are no nodes which branch off of the linear segment between $\psi_s(\sigma)$ and $\psi_s(\sigma')$. Therefore, $\varphi_e^{\psi_s(\sigma')}(0,\dots,x) \notin T_{s+1}$. Hence, R_e^σ is satisfied at the end of stage $s+1$.

This construction uses only finite injury, since only the action of R_i^τ with $i < e$ can injure R_e^σ, and R_e^σ acts at most once between times it is injured. To see that each R_e is satisfied, suppose for a contradiction that some R_e is not satisfied. Let $X \in [T]$ be such that φ_e^X is total, $\varphi_e^X \neq X$ and $\varphi_e^X \in [T]$. Fix σ of length $e+1$ such that $\psi(\sigma) \subset X$ and fix n such that $\varphi_e^X(n) \neq X(n)$. Let s be a stage such that $\psi_s(\tau)$ has reached its limit for all $|\tau| = e+1$ and all requirements of higher priority that R_e^σ have finished acting. There must be some $t > s$ and σ' with $|\sigma'| > n$ such that $\psi_t(\sigma')$ is an initial segment of X extending $\psi(\sigma)$ and $\varphi_e^{\psi_t(\sigma')}(0,\dots,m)$ is defined where m is the maximum of n and the length of $\psi(\tau)$ with $|\tau| = e+1$. R_e^σ acts at stage t, and since it is not injured by any higher priority requirements, remains satisfied forever. This contradicts our choice of X as a witness to the failure of R_e. □

Corollary 13.5. *There is an c.e. Boolean algebra B such that B has $2^{\aleph_0}$ many ultrafilters and any two distinct ultrafilters on B are Turing incomparable.*

Corollary 13.6. *There is a computable theory T such that T has $2^{\aleph_0}$ many complete consistent extensions and any two distinct complete consistent extensions are Turing incomparable.*

One of our first examples of a Π_1^0 class of sets was the class of separating sets for a fixed pair of disjoint c.e. sets. There cannot be such a Π_1^0 class that satisfies Theorem 13.4 since we can always alter a separating set on finitely many elements to obtain a different separating set. However, it turns out that this is the only restriction on proving a version of Theorem 13.4 for Π_1^0 classes of separating sets.

Theorem 13.7. *There exists disjoint c.e. sets A and B such that $A \cup B$ is coinfinite and for any two sets $C, D \in Sep(A,B)$, either $|C \triangle D| < \omega$ or C and D are Turing incomparable.*

Proof. We define a $\{0,1\}$-valued partial computable function ψ in stages by ψ_s and set $A = \{n|\psi(n) = 1\}$ and $B = \{n|\psi(n) = 0\}$. As usual, we have the stage s approximations to all of these objects. We use movable markers $x_s(i)$ to denote the i^{th} element of $\overline{A_s \cup B_s}$ at stage s. Because each $x_s(i)$ will reach a limit $x(i)$, the complement of $A \cup B$ will be infinite.

In the constructions we have seen so far, we always used strings whose domain was an initial segment of ω. Here, we sometimes use partial functions whose domain is finite but not necessarily an initial segment of ω. In particular, ψ_s is such a function.

At each stage, we let T_s be the computable tree of all $\sigma \in 2^{<\omega}$ such that σ is compatible with ψ_s. By compatible, we mean that for every $n \in \text{dom}(\psi_s)$, if $n < |\sigma|$, then $\sigma(n) = \psi_s(n)$. The Π^0_1 class $[T]$ where $T = \cap T_s$ is exactly $\text{Sep}(A, B)$.

In addition to making sure that each marker $x_s(i)$ reaches a limit (which is a simple consequence of the fact that this argument is finite injury), we need to satisfy the requirements

$$R_e : \forall X, Y \in [T](|X \bigtriangleup Y| = \omega \to \varphi_e^X \neq Y).$$

We guarantee that R_e respects the requirements R_i with $i < e$ using the set $D_{e,s} = \{x_s(i)|i < e\}$. For each subset $D \subset D_{e,s}$ we consider the subrequirement R_e^D which only works with sets X for which $X(x) = 1$ for all $x \in D$ and $X(x) = 0$ for all $x \in D_{e,s} - D$. Notice that the sets $D_{e,s}$ change with time, but they will all reach a limit at some finite stage, after which we have the final list of subrequirements R_e^D.

We define a finite partial function w associated to each R_e^D at each stage s. (This partial function w should really be subscripted by D and $D_{e,s}$, but for simplicity of notation we leave these subscripts off.) For R_e^D at stage s, w is defined with domain $D_{e,s}$ by $w(x_s(i)) = 1$ if $x_s(i) \in D$ and $w(x_s(i)) = 0$ if $x_s(i) \in D_{e,s} - D$. It is undefined for all other values. R_e^D will only look at strings which extend w, and hence it will never cause any of the markers $x_s(i)$ for $i < e$ to move. Thus, it respects the requirements of higher priority.

We say that R_e^D requires attention at stage $s+1$ if it is not currently satisfied (this term is defined during the construction) and there is a string $\mu \in T_s$ extending w (with length $\leq s$) and an m with $e \leq m \leq s$ such that $\varphi^\mu_{e,s}(x_s(m))$ converges but it is not the case that $\varphi^\mu_{e,s}(x_s(m)) = \mu(x_s(m))$. There are three ways this could happen: $x_s(m)$ is not in the domain of μ, $\varphi^\mu_e(x_s(m))$ converges to a number other than 0 or 1, or $\varphi^\mu_e(x_s(m)) = 1 - \mu(x_s(m))$.

We can now describe the construction. At stage 0, set $\psi_0 = \emptyset$ and $x_0(i) = i$ for all i. Declare all requirements currently unsatisfied. At stage $s+1$, check if there is any requirement that requires attention. If not, set $\psi_{s+1} = \psi_s$ and $x_{s+1}(i) = x_s(i)$ for all i. If there is a requirement that requires attention, then let R_e^D be the least such requirement. (We order the requirements first by e and then by a fixed computable ordering of all finite sets D.)

The action of R_e^D depends on the three possibilities listed above. First, if $x_s(m) \geq |\mu|$, then extend ψ_s to ψ_{s+1} by setting $\psi_{s+1}(x) = \mu(x)$ for all $x < |\mu|$ which are not in $\text{dom}(w)$, $\psi_{s+1}(x_s(m)) = 1 - \varphi^\mu_e(x_s(m))$, and $\psi_{s+1}(x) = \psi_s(x)$ for all other $x \in \text{dom}(\psi_s)$. Since μ is compatible with ψ_s, this definition gives $\psi_{s+1} \supset \psi_s$ and no element of $D_{e,s}$ has been added to $\text{dom}(\psi_{s+1})$. We define the new markers by setting $x_{s+1}(i)$ equal to the i^{th} element of the complement of $A_{s+1} \cup B_{s+1}$. Since no element of $D_{e,s}$ was added to $\text{dom}(\psi_{s+1})$, this leaves the markers $x_{s+1}(i) = x_s(i)$ for $i < e$.

Notice that in this case, if X is any extension of w which is in $[T_{s+1}]$, then X extends

μ. Furthermore, $\varphi_e^X(x_s(m)) = \varphi_e^\mu(x_s(m))$. However, for any Y in $[T_{s+1}]$, $Y(x_s(m)) = 1 - \varphi_e^\mu(x_s(m))$. Therefore, we have satisfied R_e^D and we declare it currently satisfied.

Second, if $\varphi_e^\mu(x_s(m)) = 1 - \mu(x_s(m))$, then we extend ψ_s to ψ_{s+1} as follows. For all $x < |\mu|$ which are not in $\text{dom}(w)$, set $\psi_{s+1}(x) = \mu(x)$, and for all other $x \in \text{dom}(\psi_s)$, set $\psi_{s+1}(x) = \psi_s(x)$. As above, we have $\psi_s \subset \psi_{s+1}$ and we have not put any element of $D_{e,s}$ into $\text{dom}(\psi_{s+1})$. We define $x_{s+1}(i)$ as above.

Consider any extension X of w in $[T_{s+1}]$ and any other set $Y \in [T_{s+1}]$. Since X is an extension of w on T_{s+1}, it must be an extension of μ. Therefore,

$$\varphi_e^X(x_s(m)) = 1 - \mu(x_s(m)) = 1 - \psi_{s+1}(x_s(m)) \neq \psi_{s+1}(x_s(m)) = Y(x_s(m)).$$

So, we have won R_e^D and declare this requirement currently satisfied.

Third, if $\varphi_e^\mu(x_s(m))$ does not have the value 0 or 1, then we extend ψ_s to ψ_{s+1} as follows. For all $x < |\mu|$ which are not in $\text{dom}(w)$, let $\psi_{s+1}(x) = \mu(x)$, and for all other $x \in \text{dom}(\psi_s)$, let $\psi_{s+1}(x) = \psi_s(x)$. We define $x_s(i)$ as before.

Consider any $X \in [T_{s+1}]$ extending w. As above, X must extend μ, and therefore, $\varphi_e^X(x_s(m))$ converges to a value other than 0 or 1. Therefore, φ_e^X does not compute a set. Again, we declare the requirement R_e^D currently satisfied.

Once we have performed the appropriate action for R_e^D, we declare all requirements of the form R_i^D with $i > e$ currently not satisfied. (These requirements have been injured in the sense that $D_{i,s+1}$ may not be equal to $D_{i,s}$, so the list of subrequirements R_i^D may have changed.) This ends the action at stage s.

It remains to verify that the construction works. Because of the finite nature of the injury, it is clear that each marker $x_s(i)$ reaches a limit $x(i)$, and hence each set $D_{e,s}$ reaches a limit D_e.

Let $X, Y \in [T]$ be sets such that $|X \triangle Y| = \omega$ and assume for a contradiction that $\varphi_e^X = Y$. Since the symmetric difference is infinite, for all n, there is an $m > n$ such that $X(x(m)) \neq Y(x(m))$. Pick a stage s and an $m > e$ such that $x_s(m) = x(m)$, $D_{e,s} = D_e$, and $X(x(m)) \neq Y(x(m))$. Fix the subset $D \subset D_e$ and the corresponding partial function w such that X extends w. Fix a stage $t > s$ such that there is an $n < t$ for which $\varphi_{e,t}^{X|n}(x(m))$ converges and no requirement of higher priority acts after stage t.

Consider the action of R_e^D as stage $t+1$. Because no requirement of higher priority acts, R_e^D is free to act if it wants to. Let $\mu = X|n$. At this stage, R_e^D sees the convergent computation $\varphi_e^\mu(x(m))$. The only thing that would prevent R_e^D from acting at this stage (and hence being satisfied forever since no higher priority requirement injures it after this stage) is if $\varphi_e^\mu(x(m)) = \mu(x(m))$. However, in this case,

$$\varphi_e^X(x(m)) = X(x(m)) \neq Y(x(m)),$$

which is a contradiction to the assumption that $\varphi_e^X = Y$. □

Another exotic type of Π_1^0 class is a thin class. Π_1^0 class of sets P is called **thin** if for every Π_1^0 class of sets $Q \subset P$, there is a clopen set U such that $Q = P \cap U$. If $P = [T]$, this is equivalent to saying that there is a finite set of nodes $\sigma_0, \ldots, \sigma_n \in T$ such that $Q = [T(\sigma_0)] \cup \cdots \cup [T(\sigma_n)]$.

Theorem 13.8. *There exists a perfect thin Π^0_1 class of sets P with no computable member.*

Proof. Let $P_e = [T_e]$ be our standard enumeration of Π^0_1 classes of sets with primitive recursive trees T_e. As in Theorem 13.4, we define P via a computable sequence of computable functions ψ_s from $2^{<\omega}$ into $2^{<\omega}$. For each ψ_s we require that

1. $\psi_s(\sigma * 0)$ and $\psi_s(\sigma * 1)$ are incompatible extends of $\psi_s(\sigma)$ for all σ,
2. $\text{range}(\psi_{s+1}) \subset \text{range}(\psi_s)$, and
3. $\lim_s \psi_s(x) = \psi(x)$ exists for all x.

Just as in Theorem 13.4, we define the computable trees T_s as the downward closure of $\text{range}(\psi_s)$ and let $P = [T]$ where $T = \cap T_s$. Therefore, P will be a perfect Π^0_1 class of sets.

To guarantee that P has no computable member, we meet the requirements

$$R_e : \varphi_e \,\text{total} \,\Rightarrow\, \forall \sigma \,\text{with}\, |\sigma| = 2e+1 (\psi(\sigma) \,\text{is incompatible with}\, \varphi_e).$$

As in the previous two theorems, we break R_e into subrequirements R_e^σ for each $|\sigma| = 2e+1$. R_e^σ tries to define $\psi_s(\sigma)$ such that $\psi_s(\sigma)$ is incompatible with φ_e. Obviously, if we meet R_e^σ for each σ, then we will have met R_e.

To insure the P is thin, we meet the requirements

$$S_e : \forall \sigma \,\text{with}\, |\sigma| = 2e+2 \left(\psi(\sigma) \in T_e \Rightarrow \forall \tau \supset \sigma (\psi(\tau) \in T_e) \right).$$

To see why the requirements S_e suffice, assume that $[T_e] \subset P$ and let U be the clopen set generated by the set of $\psi(\sigma)$ for which $|\sigma| = 2e+2$ and $\psi(\sigma) \in T_e$. Then $[T_e] = P \cap U$. We also break S_e into subrequirements S_e^σ for each $|\sigma| = 2e+2$.

At stage 0, we define $\psi_0(\sigma) = \sigma$ for all σ. At stage $2s+1$ we attempt to meet the R_e requirements. We say that R_e^σ requires attention if there is a $\tau \supset \sigma$ and an n such that $\varphi_{e,2s+1}(0, \ldots, n)$ is defined, is an extension of $\psi_{2s}(\sigma)$ and is incompatible with $\psi_{2s}(\tau)$. If no R_e^σ requires attention, then $\psi_{2s+1} = \psi_{2s}$. Otherwise, let R_e^σ be the highest priority requirement requiring attention and let τ be as above. Define $\psi_{2s+1}(\mu) = \psi_{2s}(\tau * \mu')$ if $\mu = \sigma * \mu'$ and $\psi_{2s+1}(\mu) = \psi_{2s}(\mu)$.

Notice that as in the previous theorems, ψ_{2s+1} only differs from ψ_{2s} on nodes $\mu \supset \sigma$. Therefore, R_e^σ does not interfere with any requirement R_i^ν or S_i^ν for $i < e$.

At stage $2s+2$, we attempt to meet the S_e requirements. We say that S_e^σ requires attention if $\psi_{2s+1}(\sigma) \in T_e$ and there is a $\tau \supset \sigma$ such that $\psi_{2s+1}(\tau) \notin T_e$. If there is no S_e^σ requiring attention, then $\psi_{2s+2} = \psi_{2s+1}$. Otherwise, let S_e^σ be the highest priority requirement requiring attention and let τ be as above. Define $\psi_{2s+2}(\mu) = \psi_{2s+1}(\tau * \mu')$ if $\mu = \sigma * \mu'$ and $\psi_{2s+2}(\mu) = \psi_{2s+1}(\mu)$ otherwise.

Notice that the action of S_e^σ does not interfere with the action of requirements R_i^ν with $i \leq e$ or S_i^ν with $i < e$. Therefore, this construction is also finite injury. It is an easy exercise to verify that the construction succeeds. $\square$

In order to apply Theorem 13.8 to theories and Boolean algebras, we need to examine what being thin means in these contexts.

Definition 13.9. Let Γ be a theory and let $\Gamma' \supset \Gamma$. Γ' is **finitely generated** over Γ if there are a finite number of sentence $\psi_0, \ldots, \psi_k$ such that the closed theory generated by Γ' is equal to the closed theory generated by $\Gamma \cup \{\psi_0, \ldots, \psi_k\}$.

Lemma 13.10. *Let P be a thin Π^0_1 class of sets and let Γ be the corresponding theory as in Theorem 12.5. If Γ' is any computable extension of Γ, then Γ' is finitely generated over Γ.*

Proof. For any computable extension $\Gamma' \supset \Gamma$, the corresponding Π^0_1 class of sets P' is a subclass of P. Therefore, since P is thin, P' must be equal to $P \cap U$ where U is a finite union of basic opens sets. Denote these basic open sets by $\mathcal{O}_{\sigma_0}, \ldots, \mathcal{O}_{\sigma_k}$, and as above, let $P_{\sigma_0}, \ldots, P_{\sigma_k}$ denote the corresponding sentences as in the proof of Theorem 12.5.

C_X is a complete consistent extension of Γ' if and only if $X \in P'$, which is true if and only if $X \in P \cap U$. However, $X \in P \cap U$ if and only if C_X is a complete consistent extension of $\Gamma \cup \{P_{\sigma_0}, \ldots, P_{\sigma_k}\}$. Therefore, C_X is a complete consistent extension of Γ' if and only if it is a complete consistent extension of $\Gamma \cup \{P_{\sigma_0}, \ldots, P_{\sigma_k}\}$.

By the completeness theorem, for any theory Δ and any sentence ψ, $\Delta \vdash \psi$ if and only if ψ is true in all complete consistent extensions of Δ. Therefore, by the equivalences above, $\Gamma' \vdash \psi$ if and only if $\Gamma \cup \{P_{\sigma_0}, \ldots, P_{\sigma_k}\} \vdash \psi$. In other words the closed theory generated by Γ is the same as the closed theory generated by $\Gamma \cup \{P_{\sigma_0}, \ldots, P_{\sigma_k}\}$. Therefore, Γ' is finitely generated over Γ. $\square$

Definition 13.11. A theory Γ is called **essentially undecidable** if Γ does not have any computable complete consistent extensions.

Definition 13.12. A computable essentially undecidable theory for which every computable extension is finitely generated is call a **Martin-Pour-El** theory.

Theorem 13.13. *There exists a Martin-Pour-El theory.*

Proof. Let P be the Π^0_1 class of sets in Theorem 13.8 and let Γ_P be the corresponding theory. Since P has no computable member, Γ_P is essentially undecidable. Since P is thin, every computable extension of Γ_P is finitely generated. Therefore, Γ_P is a Martin-Pour-El theory. $\square$

Finally, we show that the construction of a thin class and the construction of a class of separating sets can be combined. Notice that this theorem points out a small error in Theorem 2.30 of [5].

Theorem 13.14. *There are disjoint computably enumerable sets A and B such that $A \cup B$ is coinfinite and Sep(A, B) is thin.*

Proof. We construct a $\{0, 1\}$–valued partial computable function ψ in stages and set $A = \{n|\psi(n) = 1\}$ and $B = \{n|\psi(n) = 0\}$. To make sure that $A \cup B$ is coinfinite, the domain of ψ is coinfinite. We let ψ_s denote the portion of ψ constructed at stage s, we guarantee that $\psi_s \subset \psi_{s+1}$ and we set $\psi = \cup_s \psi_s$. We define $A_s = \{n|\psi_s(n) = 1\}$ and $B_s = \{n|\psi_s(n) = 0\}$, and we let V_s denote the set of all finite binary strings which are compatible with ψ_s. That is,

$$V_s = \{ \sigma \in 2^{<\omega} \mid \forall n < |\sigma| \, (\psi_s(n) \downarrow \to \psi_s(n) = \sigma(n)) \}.$$

Notice that $[V_s] = \mathrm{Sep}(A_s, B_s)$. Therefore, in terms of Π^0_1 classes, we construct an effective sequence of computable trees V_s such that

$$V_0 \supset V_1 \supset V_2 \supset \cdots$$

and we set $V = \cap_s V_s$. V is a Π^0_1 tree such that $[V] = \mathrm{Sep}(A, B)$.

In order to make the domain of ψ coinfinite, we use markers $\delta_s(i)$ to denote the i^{th} element of the complement of $A_s \cup B_s$ at stage s. Formally, $\delta_s(i)$ is defined so that

$$\delta_s(0) < \delta_s(1) < \delta_s(2) < \cdots$$

and $\overline{A_s \cup B_s} = \{\delta_s(i) | i \in \omega\}$. We require that $\lim_s \delta_s(i) = \delta(i)$ exists for all i. This requirement clearly makes $A \cup B$ coinfinite.

To make $\mathrm{Sep}(A, B) = [V]$ thin, we fix a standard enumeration of all the primitive recursive trees T_e and let $P_e = [T_e]$ be the e^{th} Π^0_1 class. We meet the requirements

$$R_e : \ [T_e] \subset [V] \Rightarrow \exists U (U \text{ is clopen} \wedge [T_e] = [V \cap U]).$$

We break this requirement up into subrequirements R^σ_e for each $\sigma \in V$ such that $|\sigma| = \delta(e)$. At stage s of the construction, we approximate these requirements by working with R^σ_e for each $\sigma \in V_s$ such that $|\sigma| = \delta_s(e)$. Once $\delta_s(e)$ reaches its limit $\delta(e)$, we arrive at a final list of subrequirements.

$$R^\sigma_e : \ \sigma \in T_e \ \Rightarrow \ \forall \tau \subseteq \sigma \, (\tau \in V \rightarrow \tau \in T_e)$$

Why does this make $[V]$ thin? Suppose $[T_e] \subseteq [V]$ and let $\sigma_0, \ldots, \sigma_k$ be the strings such that $\sigma_i \in V \cap T_e$ and $|\sigma_i| = \delta(e)$. Then, $[V[\sigma_i]] \subset [T_e]$ for $0 \leq i \leq k$, and for all other $\mu \in V$ with $|\mu| = \delta(e)$, $\mu \notin T_e$. Therefore, $[T_e] \subset [V]$ implies

$$[V[\sigma_0] \cup V[\sigma_1] \cup \cdots \cup V[\sigma_k]] = [T_e]$$

as required.

At stage 0, we set ψ_0 to be undefined everywhere. Assume we are at stage $s+1$ of the construction. If $\sigma \in V_s$ with $|\sigma| = \delta_s(e)$ and $\sigma \in T_e$, then we say R^σ_e needs attention if

$$\exists \tau \supset \sigma \, (\, \tau \in V_s \wedge \tau \notin T_e \,).$$

Of course, to make this condition effective to check at this stage, we only check for R^σ_e with $e \leq s$ and only for τ with $|\tau| \leq s$. Determine the requirement that gets to act by choosing R^σ_e to be the requirement that requires attention with the least value for e and (if more than one requirement with subscript e requires attention) the least value for σ under the lexicographic ordering. (If no requirement needs attention, let $\psi_{s+1} = \psi_s$.) Assume R^σ_e gets to act at stage $s+1$ and fix the corresponding τ. Define ψ_{s+1} as follows: for all i such that $|\sigma| \leq i < |\tau|$, let $\psi_{s+1}(i) = \tau(i)$, and for all other $j \in \mathrm{dom}(\psi_s)$, let $\psi_{s+1}(j) = \psi_s(j)$. Proceed to the next stage. This completes the description of the construction.

Notice what the action of R^σ_e accomplishes. All the numbers i such that $|\sigma| \leq i < |\tau|$ are now in $\mathrm{dom}(\psi_{s+1})$. Therefore, $\delta_{s+1}(e) \geq |\tau|$ and there is a string $\tau' \supseteq \tau$ which is the unique extension of σ in V_{s+1} with length $\delta_{s+1}(e)$. Hence, the requirement R^σ_e has "become" the requirement $R_e^{\tau'}$. But, $\tau' \in V_{s+1}$ and $\tau' \notin T_e$ (since $\tau \notin T_e$ and $\tau \subseteq \tau'$), so $R_e^{\tau'}$ is satisfied. We have made progress towards meeting the larger requirement R_e.

To see that the construction succeeds, we need to verify a few simple facts. First, the new function ψ_{s+1} is consistent with ψ_s in the sense that $\psi_s \subset \psi_{s+1}$. This follows because for our chosen τ we have $\tau \in V_s$ and hence τ was compatible with ψ_s. Therefore, if $|\sigma| \leq i < |\tau|$ and $\psi_s(i) \downarrow$, we have $\psi_s(i) = \tau(i)$ and so we have not changed the value of $\psi_{s+1}(i)$ at stage $s+1$.

Second, the action of R_e^σ may change the values of $\delta_{s+1}(i)$ for $i \geq e$, but for all $i < e$, $\delta_{s+1}(i) = \delta_s(i)$. Therefore, if the functions $\delta_s(i)$ for $i < e$ have reached their limits, they are not effected by the action of R_e^σ.

Third, we need to see that the action of R_e^σ does not injury other requirements of the form R_e^μ. Recall that we say R_e^μ is satisfied if $\mu \notin T_e$. Suppose that R_e^σ acts at stage $s+1$. As described above, this action causes $\delta_s(e) < \delta_{s+1}(e)$. However, for any string $\mu \in V_s$ such that $|\mu| = \delta_s(e)$, there is a unique string $\mu' \in V_{s+1}$ such that $|\mu'| = \delta_{s+1}(e)$ and $\mu \subset \mu'$. Therefore, the requirement R_e^μ becomes $R_e^{\mu'}$ and the number of these requirements does not increase. Furthermore, if R_e^μ was satisfied at stage s (by the fact that $\mu \notin T_e$), then $R_e^{\mu'}$ is satisfied at stage $s+1$ since $\mu \subset \mu'$ and so $\mu' \notin T_e$. Therefore, the action of R_e^σ only renames the other requirements of the form R_e^μ and it does not increase the number of such requirements or injure any such requirement which was already satisfied. □

14 Retraceable sets and countable thin classes

We will use the notion of a retraceable set to construct a countable thin Π_1^0 class of sets which has rank 1.

Definition 14.1. A set $A = \{a_0 < a_1 < \cdots\}$ is **retraceable** if there is a partial computable function f such that $f(a_0) = a_0$ and for all $n > 0$, $f(a_n) = a_{n-1}$. The function f is called a **retracing** function for A.

A retraceable set A has the interesting computational property that it is computable from any infinite set $B \subset A$. There are also $2^{\aleph_0}$ many such sets. To see this, consider the tree $2^{<\omega}$ and the function $l : 2^{<\omega} \to \omega$ which assigns each string its position in the lexicographic order. For any set X, consider the set $A_X = \{l(X|n) | n \in \omega\}$. The set A_X is retraceable, since from any element $l(X|n)$, we can decode the string $X|n$ and then find $X|(n-1)$ and compute $l(X|(n-1))$. (If you draw the labeled tree, the proof becomes clear.)

We will be concerned with Π_1^0 sets A which are retraceable. For these sets, we can always assume that the retracing function is total.

Lemma 14.2. *Every Π_1^0 retraceable set has a computable retracing function.*

Proof. Fix a Π_1^0 retraceable set A and let f be a retracing function for A. The issue here is that f may not be defined on every input. To define $g(n)$, we simultaneously start computing $f(n)$ and enumerating the c.e. set $\overline{A}$. Since A is contained in the domain of f, one of these two things must occur. If we see $f(n)$ halt first, then set $g(n) = f(n)$. If n is enumerated into $\overline{A}$ first, then set $g(n) = 0$. It is clear that g is total and computable. Also, for all $n \in A$, $g(n) = f(n)$, therefore g is a retracing function for A. □

We give two alternate descriptions of Π^0_1 retraceable sets which we then apply in the construction of a thin countable Π^0_1 class of sets.

Lemma 14.3. *A Π^0_1 set $A = \{a_0 < a_1 < \cdots\}$ is retraceable if and only if there is a computable function g such that for all n, $g(a_n) = n$.*

Proof. To see the forward direction, let A be a Π^0_1 retraceable set and let f be a computable retracing function for A. To define $g(n)$, we first check if $n = a_0$. If so, we set $g(n) = 0$. Otherwise, we start computing $f(n), f^2(n) = f(f(n)), \ldots, f^m(n), \ldots$ until we see an m such that $f^m(n) = a_0$ or $f^m(n) = 0$ (assuming that $a_0 \neq 0$) or $f^m(n) \geq f^{m-1}(n)$. In the first case, we set $g(n) = m$, since if $n \in A$, then $n = a_m$. In the second and third cases, we set $g(n) = 0$, since we know $n \notin A$.

To see the backward direction, assume that g has the property that $g(a_n) = n$ and we define a partial computable retracing function f for A. We nonuniformly fix $f(a_0) = a_0$ For any $a \neq a_0$ (which may or may not be in A), we calculate $g(a)$. If $g(a) = 0$, then set $f(a) = 0$. Otherwise, $g(a) = n+1$ for some n. We enumerate $\overline{A}$ (which is computably enumerable) until we see $a - (n+1)$ many elements in $\overline{A}$ which are $< a$. Let the remaining elements $< a$ be listed in increasing order as

$$b_0 < b_1 < \cdots < b_n.$$

Set $f(a) = b_n$. If $a = a_{n+1}$, then $a_i = b_i$ for $i \leq n$, so we have in fact set $f(a_{n+1}) = a_n$. (Notice that f may not be total since if $a \notin A$, then $g(a)$ may be equal to a number larger than the number of elements below a. Of course, we could modify the procedure above to make f total, but all that is necessary for the proof is that f is defined on A, which it is.) □

Lemma 14.4. *If an infinite set $A = \{a_0 < a_1 < \cdots\}$ is defined recursively by a Π^0_1 relation $Q(x, y)$ such that for all n and x, $x = a_n$ if and only if $Q(x, \langle a_0, \ldots, a_{n-1}\rangle)$, then A is Π^0_1 and retraceable.*

Proof. Since Q is a Π^0_1 relation, we can fix an approximation $Q_s(x, y)$ to $Q(x, y)$ such that $Q_{s+1}(x, y) \rightarrow Q_s(x, y)$. (To fix this approximation, think of starting with $Q_0(x, y)$ holding of all pairs of numbers and then removing pairs as we see the Π^0_1 definition of Q fail.) We also make sure that our approximation guarantees that $Q(x, y)$ holds if and only if $\forall s Q_s(x, y)$. We define the relations $R_s(n, x)$ (uniformly in n) by

$$R_s(n,x) \Leftrightarrow \exists x_0, \ldots, x_{n-1}\big(x_0 < x_1 < \cdots < x_{n-1} < x \wedge Q_s(x, \langle x_0, \ldots, x_{n-1}\rangle) \wedge \forall i < n(R_s(i, x_i))\big).$$

We claim that for all n, $x = a_n$ if and only if $\forall s(R_s(n, x))$. We prove this claim by induction on n. Assume that $n = 0$. In this case, $R_s(0, x)$ holds if and only if $Q_s(0, \lambda)$ holds. Universally quantifying over s gives the first of the following equivalences.

$$\forall s(R_s(0, x)) \Leftrightarrow \forall s(Q_s(x, \lambda)) \Leftrightarrow Q(x, \lambda) \Leftrightarrow x = a_0$$

The second equivalence follows from the definition of our approximation to Q by Q_s, and the third equivalence follows from the fact that Q defines the sequence of elements a_i.

Next assume that this equivalence holds for $i < n$ and we prove it for n. To show the $(\Rightarrow)$ direction, let $x_i = a_i$ for $i < n$. We know by induction that $\forall s R_s(i, a_i)$ and by the definition of Q that $Q(a_n, \langle a_0, \dots, a_{n-1} \rangle)$, so $\forall s Q_s(a_n, \langle a_0, \dots, a_{n-1} \rangle)$. Therefore, the a_i give the appropriate witnesses for $\forall s R_s(n, a_n)$.

To show the ($\Leftarrow$) direction, chose s large enough that

$$\forall i < n \, \forall y < x \, \big(R_s(i, y) \leftrightarrow y = a_i\big).$$

That is, by induction, for all $i < n$ and all $y < x$, if $y \neq a_i$, there is a stage u at which we see $\neg R_u(i, y)$. We take the maximum over the appropriate stages to get the stage s desired above. For any $t > s$, the only witness x_i such that $R_t(i, x_i)$ holds is $x_i = a_i$. Therefore, $R_t(n, x)$ holds if and only if $Q_t(x, \langle a_0, \dots, a_{n_1} \rangle)$. By the definition of Q, we know that the unique x such that $\forall t > s \, Q_t(x, \langle a_0, \dots, a_{n-1} \rangle)$ is $x = a_n$, as required.

Now that both directions of our claim have been established, we show that A is both Π^0_1 and retraceable. A is Π^0_1 since

$$a \in A \Leftrightarrow \exists n \leq a \, \forall s \, R_s(n, a).$$

To see that f is retraceable, we define a computable g such that $g(a_n) = n$. Given any a, we know that there is at most one $n \leq a$ such that $a = a_n$. In other words, there is at most on $n \leq a$ such that $\forall s R_s(n, a)$. Calculate $R_s(n, a)$ for each $n \leq a$ and for increasing values of s until there is only one value n left such that $R_s(n, a)$. Set $g(a) = n$ for this n. □

Definition 14.5. For any set $A = \{a_0 < a_1 < \cdots\}$, let $P(A)$ be the set of all initial segments of A (as subsets of ω):

$$P(A) = \{\, X \mid X = A \vee X = \emptyset \vee \exists n \, (X = \{a_0, \dots, a_n\}) \,\}.$$

The proof of the next lemma follows directly from the appropriate definitions.

Lemma 14.6. *For any set A, there is a tree $T_A \subset 2^{<\omega}$ such that $T_A \leq_T A$ and $[T_A] = P(A)$.*

Theorem 14.7. *If A is a Π^0_1 retraceable set, then $P(A)$ is a Π^0_1 class of sets.*

Proof. For any sequence $\sigma \in 2^{<\omega}$, let $b_0 < b_1 < \cdots < b_k < |\sigma|$ be all the numbers such that $\sigma(b_i) = 1$. We define σ^* to be the sequence $\langle b_0, \dots, b_k \rangle$. Since A is Π^0_1, we assume we have an approximation A_s to A such that

$$\omega = A_0 \supset A_1 \supset A_2 \supset \cdots$$

with $A = \cap A_s$. Fix a computable retracing function f for A.

We define a computable tree T such that $P(A) = [T]$. Let σ be a string such that $|\sigma| = s$ and $\sigma^* = \langle b_0, \dots, b_k \rangle$. Let

$$\sigma \in T \Leftrightarrow \forall i \leq k \big(b_0 = a_0 \wedge b_i \in A_s \wedge (i > 0 \rightarrow f(b_i) = b_{i-1})\big).$$

If $\{b_0, \dots, b_k\} \in P(A)$, then for all τ such that $\tau^* = \sigma^*$, $\tau \in T$. Therefore, the set $\{b_0, \dots, b_k\} \in [T]$ as required. (We leave it for the reader to check that both $\emptyset$ and A are in $[T]$.)

If σ^* is not an initial segment of A, then we must have one of the following two situations.

- Some $b_i \notin A$. In this case, for large enough s, $b_i \notin A_s$. Hence, for all τ with $|\tau| > s$ and $\tau^* = \sigma^*$, $\tau \notin T$. Therefore, σ is not the initial segment of a set in $[T]$.
- $\{b_0, \ldots, b_k\} \subset A$ but is not an initial segment of A. Then either $b_0 \neq a_0$ or, by the properties of a retracing function, there is some b_{i+1} such that $f(b_{i+1}) \neq b_i$. In either case, $\sigma \notin T$.

□

We now arrive at the main theorem of this section.

Theorem 14.8. *There is a countable thin Π^0_1 class.*

Proof. We build a Π^0_1 retraceable set A such that $P(A)$ is a thin Π^0_1 class. We first motivate the requirements. Suppose $A = \{a_0 < a_1 < \cdots\}$ and let $A_n = \{a_i | i < n\}$ for each n. We let T_e be our fixed enumeration of all primitive recursive trees. To make $P(A)$ thin, we meet

$$R_e : A \in [T_e] \Rightarrow \forall n \geq e(A_n \in [T_e]).$$

To see why this makes $P(A)$ thin, assume $[T_e] \subset P(A)$ and split into two cases.

First, if $A \notin [T_e]$, then $|[T_e]| = |[T_e] \cap P(A)| < \omega$ since A is the only nonisolated point in $P(A)$. Therefore, $[T_e]$ consists of a finite number of isolated points and for the appropriate choice of a clopen U, $[T_e] = P(A) \cap U$.

Second, if $A \in [T_e]$, then by R_e,

$$\big(\{A\} \cup \{A_n | n \geq e\}\big) \subseteq [T_e].$$

Therefore, $|P(A) \setminus [T_e]| < \omega$. Let $\sigma \subset A$ be such that $\sigma^* = A_e$. Then, $(P(A) \cap [\sigma]) \subseteq [T_e]$ and there are only finitely many paths in $[T_e]$ which are not in $P(A) \cap [\sigma]$. If V is a clopen set covering these paths, then $(P(A) \cap (V \cup [\sigma])) = [T_e]$. This completes the proof that meeting the requirements is sufficient to make $P(A)$ thin. Therefore, it suffices to build a Π^0_1 retraceable set which meets requirements R_e.

We define A using Lemma 14.4. To define a_n, assume that we have already determined $a_0, \ldots, a_{n-1}$. Let $Q^*(a, \langle a_0, \ldots, a_{n-1}\rangle)$ be the relation which holds if and only if the following two properties hold for all $m < n$:

- $a_m < a$ and
- either the unique string σ with $|\sigma| = a$ and $\sigma^* = \{a_0, \ldots, a_{n-1}\}$ is not in T_m, or for all $y \geq a$, the unique string τ with $|\tau| = y$ and $\tau^* = \{a_0, \ldots, a_{n-1}\}$ is in T_m.

It is clear that Q^* is a Π^0_1 relation. We let a_n be the least a such that $Q^*(a, \langle a_0, \ldots, a_{n-1}\rangle)$ holds. More formally, let $Q(a, \langle a_0, \ldots, a_{n-1}\rangle)$ hold if and only if Q^* holds on these numbers and for all $x < a$, either $x \leq a_{n-1}$ or there is an $m < n$ such that the unique string σ with $|\sigma| = x$ and $\sigma^* = \{a_0, \ldots, a_{n-1}\}$ is in T_m and the unique string τ with $|\tau| = a$ and $\tau^* = \{a_0, \ldots, a_{n-1}\}$ is not in T_m. We then set $a_n = a$ if and only if $Q(a, \langle a_0, \ldots, a_{n-1}\rangle)$.

It remains to check the required properties. A is Π^0_1 and retraceable by Lemma 14.4. To see that $P(A)$ satisfies R_e, suppose $A \in [T_e]$ and fix $n \geq e$. We show that $A_n \in [T_e]$. Let σ be the unique string such that $|\sigma| = a_n$ and $\sigma^* = \{a_0, \ldots, a_{n-1}\}$. The fact that $A \in [T_e]$ implies that $\sigma \in T_e$. But, by the definition of Q, we could only have set a_n to be this particular value if for all $x \geq a_n$, the unique string τ with $|\tau| = x$ and $\tau^* = \{a_0, \ldots, a_{n-1}\}$ is in T_e. This statement exactly says that $A_n \in [T_e]$ as required. □

Notice that the proof of Theorem 14.8 does more than just construct a countable thin Π^0_1 class P. It also forces P to have a unique nonisolated path, which is exactly the set A. It is not hard to modify the proof above to make $A \equiv_T 0'$. To do this, we add another requirement

$$S_e : e \in 0' \Leftrightarrow e \in 0'_{a_e}.$$

To meet these requirements, we add the following clause to the definition of $Q^*(a, \langle a_0, \ldots, a_{n-1}\rangle)$: $n \in 0' \rightarrow n \in 0'_a$. Writing this clause without the implication gives $n \notin 0' \vee n \in 0'_a$, which is clearly Π^0_1. We therefore arrive at the following corollary.

Corollary 14.9. *There is a computable theory T such that T has countably many complete consistent extensions, all of which are computable except one which is $\equiv_T 0'$. Furthermore, every computable extension is finitely generated over T.*

We end this section with a result which states that every Π^0_1 retraceable set A can be realized as the unique nonisolated path through a computable tree in which there are no deadends. Recall that the Cantor-Bendixson derivative CB($[T]$) of a binary branching tree T is the set of all nonisolated paths in $[T]$.

Lemma 14.10. *For any Π^0_1 retraceable set A, there is a computable binary branching tree T for which Ext(T) $= T$ and CB($[T]$) $= \{A\}$.*

Proof. Fix a computable approximation A_s to A such that $A_{s+1} \subset A_s$ and $A = \cap A_s$. (Since A is a Π^0_1 set, we think of starting with $A_0 = \omega$ and eliminating elements from A_{s+1} if they are enumerated into the c.e. set $\overline{A}$.) Let f be a computable retracing function for A and let a_0 denote the least element of A.

To any string $\sigma \in 2^{<\omega}$, we associate the finite set $\{n < |\sigma| \,|\, \sigma(n) = 1\}$ and we let σ^* be this set listed as an increasing sequence $\langle b_0, \ldots, b_k\rangle$. The idea of this proof is to put σ into T if it looks like σ^* is an initial segment of A. To guarantee that every node is extendible to a path, for every node $\sigma \in T$, we put $\sigma * 0$ into T.

We define T in stages. At stage s, we define T_s which consists of all strings of length $\leq s$ which will be in T. Let σ be the unique sequence of length $a_0 + 1$ which contains all zeros except for the last entry which is a 1. Therefore, $\sigma^* = \langle a_0\rangle$, which we know is an initial segment of A. We define $T_{a_0+1} = \{\sigma\}$ and continue to build T at higher levels by induction.

Assume that T_s has been defined for $s > a_0$. We assume two induction hypotheses.

1. For each $\sigma \in T_s$ of length s, if $\sigma^* = \langle b_0, \ldots, b_k\rangle$, then each $b_i \in A_{b_k}$, $b_0 = a_0$ and $f(b_i) = b_{i-1}$ for $i > 0$.

2. If $|\sigma| \leq s$, $\sigma \notin T_s$, and $\sigma^* = \langle b_0, \ldots, b_k\rangle$, then either $b_0 \neq a_0$ or $f(b_i) \neq b_{i-1}$ for some $i > 0$, or at stage $|\sigma|$ we saw that some $b_i \notin A$.

By condition (1), each σ^* looks like a possible initial segment of A. By condition (2), we have only eliminated strings σ such that σ^* is definitely not an initial segment of A.

We define T_{s+1} in a sequence of steps.

1. Put $\sigma * 0$ into T_{s+1} for all $\sigma \in T_s$ of length s.

2. Check if $s \in A_s$. If not, then end the definition of T_{s+1} here.

3. Otherwise, calculate $f(s)$. Check if there is any $\sigma \in T_s$ of length s such that $f(s)$ is equal to the largest number in σ^*. If there is no such string, then end the definition of T_{s+1} here.

4. If there is such a string, then by the first induction hypothesis, there is a unique such string. Let σ denote this string and let $\tau = \sigma * 1$. Assume $\tau = \langle b_0, \ldots, b_k, s \rangle$. τ is the lone remaining candidate for entry into T_{s+1}. Check if each $b_i \in A_s$. If not, then do not put τ into T_{s+1} and end the definition of T_{s+1} here.

5. Otherwise, for each string $\delta \in T_s$ of length s, check if $|\delta^*| > k+1$ and $\delta^*|(k+1) = \tau^*|k+1$. That is, check if $\delta^* = \langle b_0, \ldots, b_k, c_{k+1}, \ldots, c_l \rangle$. For each such δ, we know that it is not the case that both δ^* and τ^* are initial segments of A. In particular, either $s \notin A$ or $c_{k+1} \notin A$. Enumerate $\overline{A}$ until we discover which element is not in A. If $s \notin A$, then do not put τ into T_{s+1} and end the definition of T_{s+1}. Otherwise, compare τ with the next string that meets the criteria for δ.

6. If we have compared τ with all δ meeting these criteria and we still believe that $s \in A$, then put τ into T_{s+1}.

This ends the definition of T_{s+1}. We need to check that the induction hypotheses are satisfied. For each string of the form $\sigma * 0$ in T_{s+1} of length $s+1$, the first induction hypothesis is met trivially since $\sigma * 0^* = \sigma^*$. If the unique string τ was enumerated into T_{s+1}, then it also meets the first induction hypothesis because of the checks in steps (2) and (3).

For the second induction hypothesis, fix any string γ of length $s+1$ which is not in T_{s+1} and let γ' be the predecessor of γ. If γ' is not in T_s, then we meet the second induction hypothesis for γ because γ' met the second induction hypothesis at stage s. If $\gamma' \in T_s$, then $\gamma = \gamma' * 1$. If $\gamma \neq \tau$, then we know $f(s)$ is not equal to the largest element of γ'^* and hence we meet the second hypothesis. If $\gamma = \tau$, then we must have seen that $s \notin A$ or else we would have put γ into T_{s+1}. Therefore, both induction hypotheses are met.

It remains check that T has the required properties. Since $\sigma \in T$ implies $\sigma * 0 \in T$, T has no terminal nodes.

We show next that $A \in [T]$ and $A \in \mathrm{CB}([T])$. Fix $n > 0$ and let σ_n be the unique string of length $a_n + 1$ such that $\sigma_n^* = A|n$. We show by induction on n that $\sigma_n \in T$. At stage $a_0 + 1$, we put σ_1 into T. Assume that the result holds for numbers $\leq n$. Since $\sigma_n \in T_{a_n+1}$, at stage $a_{n+1} + 1$, there is a unique node γ of length a_{n+1} which is the extension of σ_n by all zeros. Since $a_{n+1} \in A$ and $f(a_{n+1}) = a_n$, we set $\tau = \gamma * 1 = \sigma_{n+1}$ at stage $a_{n+1} + 1$. Because every element of $\sigma_{n+1}^* \in A$, we do not terminate the definition of T_{s+1} until we put σ_{n+1} into T_{s+1}.

Therefore, each string $A|n$ is in T. This fact immediately implies that $A \in [T]$ and that for every m, $A|n * 0^m \in T$. Therefore, A is not isolated in $[T]$, so $A \in \mathrm{CB}([T])$.

Finally, suppose that $X \neq A$ and $X \in [T]$. We need to show that X is isolated in T. There are two cases to consider. If there is an $n \in X \setminus A$, then let $s > n$ be such that $n \notin A_s$. Let $\sigma = X|s$. Since $n \in \sigma^*$ and $n \notin A_s$, for every extension τ of σ in T, $\tau(x) = 0$ for all $x \geq s$. Therefore, X is isolated by the open set $\mathcal{O}_\sigma$.

Otherwise, $X \subset A$, but there is a least $a_n \notin A$. However, the only element of $a \in A$ for which $f(a) = a_{n-1}$ is $a = a_n$. Therefore, $A = \{a_0, \ldots, a_{n-1}\}$. Let σ be the node of length $a_n + 1$ which extends σ by all zeros. At stage $a_n + 1$, we put σ_n into T. We claim that the only extensions of σ that enter T after this stage are extensions by zeros. Suppose for a contradiction that at some later stage $s + 1$ the designated τ node is an extension of σ. Then $\tau(s) = a_{n-1}$. Since $s \neq a_n$, we will compare τ with an extension of σ_n and wait for either s or a_n to be enumerated into $\overline{A}$. Since $a_n \in A$, we must see $s \in \overline{A}$ and hence we do not put τ into T. Therefore, the only extensions of σ in T are extensions by zeros, so X is isolated by $\mathcal{O}_\sigma$. □

15 Measure and category

In this section, we discuss several of the basic results on measure and category as they relate to naturally defined subsets of 2^ω in recursion theory. We begin with the category results, since these are typically simpler.

Definition 15.1. A class of sets $P \subset 2^\omega$ is **dense in the basic open set $\mathcal{O}_\sigma$** if for every string τ, $P \cap \mathcal{O}_{\sigma * \tau} \neq \emptyset$. P is **dense** if it is dense in $\mathcal{O}_\lambda$. P is **nowhere dense** if it is not dense in any basic open set. P is **meager** (or **of first category**) if it is a countable union of nowhere dense sets. P is **nonmeager** (or **of second category**) if it is not meager and is **comeager** if its complement is meager.

The intuition is that a meager set denotes a set which is some sense "small". The simplest example of a nowhere dense set P is a singleton $P = \{A\}$. Since any countable set is the countable union of singleton sets, any countable set is meager. Furthermore, it is easy to check that any finite or countable union of meager sets is meager and that any subset of a meager set is meager. The following lemma presents the standard example of a nonmeager set.

Lemma 15.2. *No basic open set $\mathcal{O}_\sigma$ is meager.*

Proof. Assume that $\mathcal{O}_\sigma$ is meager and is equal to $\cup_{n \in \omega} A_n$ where each A_n is nowhere dense. To derive a contradiction, we produce a set Y such that $Y \in \mathcal{O}_\sigma$, but $Y \notin A_n$ for all n.

We define a sequence of strings τ_n such that

$$(A_0 \cup \cdots \cup A_n) \cap \mathcal{O}_{\sigma * \tau_0 * \tau_1 * \cdots * \tau_n} = \emptyset.$$

We let Y be the limit of the sequence σ, $\sigma * \tau_0$, $\sigma * \tau_0 * \tau_1$, and so on. Then $Y \in \mathcal{O}_\sigma$ because $\sigma \subset Y$ and $Y \notin A_n$ because of the intersection property above.

To define τ_0, we use the property that A_0 is nowhere dense. In particular, A_0 is not dense in $\mathcal{O}_\sigma$, so there is a string τ_0 such that $A_0 \cap \mathcal{O}_{\sigma*\tau_0} = \emptyset$. We proceed by induction. Assume that τ_n has been defined with the intersection property above. Since A_{n+1} is nowhere dense, it is not dense in $\mathcal{O}_{\sigma*\tau_0*\cdots*\tau_n}$. Therefore, there is a τ_{n+1} such that $A_{n+1} \cap \mathcal{O}_{\sigma*\tau_0*\cdots*\tau_{n+1}} = \emptyset$. Combining this equality with the intersection property above yields the desired induction hypothesis. □

Theorem 15.3. *The following statements all hold and are each referred to as the Baire Category Theorem.*

- *No nonempty open set is meager.*
- *The complement of a meager set is dense.*
- *The intersection of countably many dense open sets is dense.*

Proof. To see (1), recall that any open set contains a basic open set. Therefore, (1) follows from Lemma 15.2 and the fact that any subset of a meager set is meager . To see (2), assume that P is meager. Since any subset of a meager set is meager, we know by Lemma 15.2 that $\mathcal{O}_\sigma \not\subset P$ for all σ. Hence, $\mathcal{O}_\sigma \cap \overline{P} \neq \emptyset$ for all σ. Therefore, $\overline{P}$ is dense.

To see (3), let P_n be dense open sets and P be the intersection of these sets. Fix any n and any σ. Since P_n is dense, $P_n \cap \mathcal{O}_\sigma \neq \emptyset$. Since P_n is open, this intersection is also open, so it must contain a basic open set. Therefore, there is a $\tau \supset \sigma$ with $\mathcal{O}_\tau \subset P_n$. In other words, for all σ, there is a $\tau \supset \sigma$ such that $\overline{P_n} \cap \mathcal{O}_\tau = \emptyset$. Therefore, each $\overline{P_n}$ is nowhere dense. Since P is the complement of $\cup\overline{P_n}$, (3) follows from (2). □

We can now prove our first result concerning category.

Theorem 15.4. *For any noncomputable set X, $U_X = \{Y | X \leq_T Y\}$ is meager.*

Proof. Let $U_e = \{Y | X = \varphi_e^Y\}$. Assume for a contradiction that U is not meager. Since U is the countable union of the U_e sets, one of these sets must not be nowhere dense. Fix e and σ such that U_e is dense in $\mathcal{O}_\sigma$. We derive a contradiction by giving a computable procedure to compute X. We claim that

$$X(m) = n \Leftrightarrow \exists \tau \supset \sigma \exists s (\varphi_{e,s}^\tau(m) = n).$$

We can search for appropriate τ and s, taking the first pair that we come across. Since the characteristic function of X is total, we know that we will find such a pair. Hence, once we verify this equivalence, we will have the desired contradiction.

To see the $\Rightarrow$ direction, we use the denseness assumption. U_e is dense in $\mathcal{O}_\sigma$, so there is a $Y \in U_e$ with $\sigma \subset Y$. Taking an appropriate extension $\sigma \subset \tau \subset Y$ and stage s, we get the desired computation.

To see the $\Leftarrow$ direction, fix τ and s. For a contradiction, assume that $\varphi_{e,s}^\tau(m) \neq X(m)$. Then for any $Y \in \mathcal{O}_\tau$, $\varphi_e^Y \neq X$. Hence, $U_e \cap \mathcal{O}_\tau = \emptyset$. Since $\sigma \subset \tau$, this contradicts the assumption that U_e is dense in $\mathcal{O}_\sigma$. □

We mention one application of this theorem to classical recursion theory which utilizes the Baire Category Theorem.

Corollary 15.5. *For any noncomputable X, there are uncountably many Y such that X and Y are Turing incomparable.*

Proof. Let Q_X be the class of all sets which are Turing incomparable with X and let P be the class of sets which are computable from X. Since P is countable, it is meager. Since the union of meager sets is meager, $P \cup U_X$ (where U_X is as in Theorem 15.4) is meager. Since $Q_X = \overline{P \cup U_X}$, Q_X is comeager. The Baire Category Theorem says that any comeager set is dense, and hence it is not meager. Therefore, Q_X is not meager. In particular, this means that Q_X is not countable. □

Corollary 15.6. *For any noncomputable set X, Q_X is dense.*

Proof. In the previous corollary, we shown that for any noncomputable set X, Q_X is comeager. By the Baire Category Theorem, any comeager set is dense. □

We can alter the proof of Theorem 15.4 very slightly to obtain a similar result for Π^0_1 classes of sets.

Theorem 15.7. *Let P be a Π^0_1 class of sets with no computable member and let U_P be the class of all sets which can compute some member of P. Then U_P is meager.*

Proof. Let U_e denote the class of sets Y such that $\varphi_e^Y \in P$. Assume that U_P is not meager. As above, we can fix e and σ such that U_e is dense in $\mathcal{O}_\sigma$. This means that any string $\tau \subset \sigma$ can be extended to a set in U_e.

Fix a computable tree T such that $P = [T]$. To derive a contradiction, we construct a computable set $X \in P$. First, notice that if τ extends σ and $\varphi_e^\tau(0), \dots, \varphi_e^\tau(n)$ converge, then the string $\langle \varphi_e^\tau(0), \dots, \varphi_e^\sigma(n)\rangle$ must be in T. If it were not in T, then τ could not be extended to a set $Y \in U_e$. Second, notice that there also must be an extension $\tau' \supset \tau$ such that $\varphi_e^{\tau'}(n+1)$ converges. Otherwise, τ could not be extended to a set Y such that φ_e^Y is total (much less equal to X).

To construct the computable set X, look for strings τ_n such that $\sigma \subset \tau_0 \subset \tau_1 \subset \cdots$ and for all n, $\varphi_e^{\tau_n}(n)$ converges. By the comments above, such strings can be found by searching and $\langle \varphi_e^{\tau_0}(0), \dots, \varphi_e^{\tau_n}(n)\rangle \in T$ for all n. Set

$$X|(n+1) = \varphi_e^{\tau_0}(0), \dots, \varphi_e^{\tau_n}(n)\rangle.$$

Then X is a computable set on P, yielding the desired contradiction. □

We now turn to some basic measure theoretic results. Recall that we assign the product measure μ of the "fair coin flip" measure on $\{0,1\}$ to 2^ω. That is, we assign measure $1/2$ to each of the subsets $\{0\}$ and $\{1\}$ in $\{0,1\}$ and then use the product measure on $\{0,1\}^\omega$. Under this measure, the set $[2^{<\omega}(\sigma)]$ has measure 2^{-n} where $n = |\sigma|$.

We will only use a few facts about this measure from general measure theory, and we present them here as a reminder. Any facts which we use that are specific to this measure (for example Lemma 15.8), we will prove.

- μ is defined on all Borel sets in 2^ω.
- μ is countably additive.
- For any measurable set X and any $\epsilon > 0$, there is an open set A such that $X \subset A$ and $\mu(A) < \mu(X) + \epsilon$.

The next lemma says that the measure of any open set G can be approximated to any specified degree by the measure of a finite union of basic open sets contained in G.

Lemma 15.8. *Let G be any open set in 2^ω and let $\epsilon > 0$ be arbitrary. There are basic open sets defined by $\sigma_0, \dots, \sigma_k$ which are each contained in G and such that*

$$\mu(G) < 2^{-|\sigma_0|} + \cdots + 2^{-|\sigma_k|} + \epsilon.$$

Proof. Since G is an open set in 2^ω, $\overline{G}$ is a closed set which can be represented by $[T]$ for some binary branching tree T. Since we are not concerned with recursion theoretic issues here, we can assume that T has not nonextendible nodes. Recall that $\mathcal{O}_\sigma$ is the basic open set consisting of all sets extending σ. G is equal to the countable union of all $\mathcal{O}_\sigma$ such that $\sigma \notin T$. Pictorially, we can think of G as represented by a countable union of cones with bases $\sigma \notin T$. We thin this set of cones out by removing the cones which lie above other cones. Let $\sigma_0, \sigma_1, \dots$ be the strings such that $\mathcal{O}_{\sigma_i} \subset G$ and $\forall \tau \subsetneq \sigma_i(\mathcal{O}_\tau \not\subset G$. These cones give a countable cover of G by disjoint basic open sets.

If this sequence of cones is finite, then G is equal to a finite union of disjoint basic open sets and we are done. Otherwise, by countable additivity,

$$\mu(G) = \Sigma_{i\in\omega} 2^{-|\sigma_i|}.$$

Since this sum converges, it can be approximated by finite initial segments to any specified degree of accuracy. If k is such that $\Sigma_{i>k} 2^{-|\sigma_i|} < \epsilon$, then $\sigma_0, \dots, \sigma_k$ are the required basic open sets. □

Theorem 15.9. *For any noncomputable set X, the set $U_X = \{Y | X \leq_T Y\}$ has measure zero.*

Proof. The set U_X is the countable union of $U_{e,X}$ for $e \in \omega$ where $U_{e,X} = \{Y | X = \varphi_e^Y\}$. Because μ is countably additive, it suffices to show $\mu(U_{e,X}) = 0$ for all e.

First, we show that each $U_{e,X}$ is Borel and hence measurable. Let $U_{e,n,X}$ denote the set of all Y such that $\exists s(\varphi_{e,s}^Y(n) = X(n))$. $U_{e,n,X}$ is the complement of the Π_1^0 class of all sets Z such that $\forall s(\varphi_{e,s}^Z(n) \neq X(n))$. Therefore, not only is $U_{e,n,X}$ an open set, it is an effectively open set. (Formally, it is a Σ_1^0 class, being the complement of a Π_1^0 class.) Since

$$Y \in U_{e,X} \Leftrightarrow \forall n \exists s(\varphi_{e,s}(n) = X(n)),$$

$U_{e,X}$ is the intersection of $U_{e,n,X}$ for all n. Therefore, $U_{e,X}$ is a G_δ set and hence Borel.

Now that we know $U_{e,X}$ is measurable, we fix e and assume for a contradiction that $\mu(U_{e,X}) = 4m > 0$. We show that this assumption implies that X is computable, contradicting the hypothesis of the theorem.

Since $U_{e,X}$ is measurable, so is $\overline{U_{e,X}}$. Let G be an open set containing $\overline{U_{e,X}}$ such that $\mu(G) < \mu(\overline{U_{e,X}}) + m$. Notice that $\overline{G} \subset U_{e,X}$ and $\mu(\overline{G}) > 3m$. By Lemma 15.8, we can approximate G by basic open sets defined by $\sigma_0, \ldots, \sigma_k$ within measure m. Therefore, if B is the union of these basic open sets, then $B \subset G$ and $\mu(G) < \mu(B) + m$. Notice that since $\overline{U_{e,X}} \subset G$, we have

$$\mu(\overline{B} \cap \overline{U_{e,X}}) < \mu(\overline{B} \cap G) < m.$$

Let $S_{i,n}$ be the set of all $Z \in \overline{B}$ such that $\varphi_e^Z(n) = i$. By the arguments similar to those given above, each $S_{i,n}$ is Borel and hence measurable. If $i = X(n)$, then $\overline{G} \subset S_{i,n}$ and so $\mu(S_{i,n}) \geq 3m$. On the other hand, if $i \neq X(n)$, then $S_{i,n} \subset \overline{B} \cap \overline{U_{e,X}}$ and hence $\mu(S_{i,n}) \leq m$.

To arrive at our contradiction, we give a computable procedure for computing X. Since the list of strings $\sigma_0, \ldots, \sigma_k$ defining the open set B is finite, we can assume that we are given those strings.

Fix n and we show how to compute $X(n)$. For any string τ we can check if $\varphi_{e,|\tau|}^{\tau}(n)$ converges. If so, then let i be such that for all $Z \in \mathcal{O}_\tau$, $\varphi_e^Z(n) = i$. Since both $\mathcal{O}_\tau$ and B are finite unions of basic open sets and we know the strings defining those basic open sets, we can effectively determine a finite sequence of strings which define pairwise disjoint basic open sets whose finite union is equal to $\mathcal{O}_\tau - B$.

Let $D_\tau = \mathcal{O}_\tau - B$. Because we have a finite description of D_τ as a union of pairwise disjoint basic open sets, we can effectively determine $\mu(D_\tau)$. Since $D_\tau \subset \overline{B}$ and for all $Z \in D_\tau$, $\varphi_e^Z(n) = i$, we have that $D_\tau \subset S_{i,n}$. Therefore, we know that $\mu(S_{i,n}) \geq \mu(D_\tau)$.

By considering the strings τ in $2^{<\omega}$ in lexicographic order, we can determine more and more sets D_τ. These sets give us better and better approximations to $\mu(S_{i,n})$. Since $S_{X(n),n}$ is the only set of the form $S_{i,n}$ with measure $> m$, we continue to determine sets D_τ until we find some $S_{i,n}$ with $\mu(S_{i,n}) \geq 2m$. At this point, we know that $X(n) = i$. Therefore, X is computable which contradicts the hypothesis of the theorem. □

Theorem 15.10. *Let A and B be disjoint computablely inseparable c.e. sets and let $U = \{X | \exists Y \in Sep(A,B)(Y \leq_T X)\}$. Then, $\mu(U) = 0$.*

16 Comments on references

I would like to make a small, completely inadequate, number of comments on the references for this material. The general presentation in Sections 1, 2 and 4 is very similar to that of [1] and [2]. For a similar (and much more in depth) presentation of the material from Sections 5 and 6, see [6] and [7]. The coding in Sections 7 and 12 as well as the equivalence of the definitions in Section 9 can be found in [5] and [3]. These articles also contain many more applications of Π_1^0 classes to effective mathematics. Several of the lemmas in Section 10 come from [16] and this paper contains many nice results on applications of Π_1^0 classes that are beyond the scope of this course (unfortunately). The results of Sections 11 and 13 come from the seminal papers [11] and [12]. These papers contain quite a number of other similar results as well as nice applcations of these results to theories. The material for Section 14 comes from [4] which, as with many of the articles cited above, goes into far more detail about countable

Π^0_1 classes than we have time for in this course. The work in Section 15 can be found either in [11], [12] or [14].

References

[1] J.L. Bell & M. Machover, A Course in Mathematical Logic, North-Holland, Amsterdam 1977.

[2] J.L. Bell & A.B. Slomson, Models and Ultraproducts: An Introduction, North-Holland, Amsterdam 1969.

[3] D. Cenzer, Π^0_1 classes, in [10], 37-85.

[4] D. Cenzer, R.G. Downey, C.G. Jockusch Jr. & R.A. Shore, Countable thin Π^0_1 classes, *Annals of Pure and Applied Logic* **59** (1993), 79-139.

[5] D. Cenzer & J.B. Remmel, Π^0_1 classes in mathematics, in volume 2 of [8], 623-821.

[6] R.G. Downey, On presentations of algebraic structures, in Complexity, Logic and Recursion Theory (ed. Sorbi), Lecture Notes in Pure and Applied Mathematics **187** (1997) 157-205.

[7] R.G. Downey, Computability theory and linear orderings, in [8], 823-976.

[8] Y.L. Ershov, S.S. Goncharov, A. Nerode & J.B. Remmel (ed.), Handbook of Recursive Mathematics (2 Volumes), Elsevier, Amsterdam, 1998.

[9] S.S. Goncharov, Countable Boolean Algebras and Decidability, Kluwer New York 1997.

[10] E.R. Griffor, Handbook of Recursion Theory, Elsevier, Amsterdam 1999.

[11] C.G. Jockusch Jr. & R.I. Soare, Degrees of members of Π^0_1 classes, *Pacific Journal of Mathematics* **40** (1972), 605-616.

[12] C.G. Jockusch Jr. & R.I. Soare, Π^0_1 classes and degrees of theories, *Transactions of the American Mathematical Society* **173** (1972), 33-56.

[13] H. Rogers Jr., Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York 1967.

[14] G.E. Sacks, Degrees of Unsolvability, Princeton University Press, Princeton, NJ 1993.

[15] S.G. Simpson, Subsystems of Second Order Arithmetic, Springer-Verlag, Berlin 1999.

[16] S.G. Simpson, Π^0_1 sets and models of WKL_0, to appear in Reverse Mathematics 2001.

[17] R.I. Soare, Recursively Enumerable Sets and Degrees, Springer-Verlag, Berlin 1987.