5-8-2015

Computability Theory and Ordered Groups

Caleb J. Martin
University of Connecticut - Storrs, martin.caleb.j@gmail.com

Computability Theory and Ordered Groups

Caleb J. Martin, Ph.D.

University of Connecticut, 2015

ABSTRACT

Ordered abelian groups are studied from the viewpoint of computability theory. In particular, we examine the possible complexity of orders on a computable abelian group. The space of orders on such a group may be represented in a natural way as a Π^0_1 class, but not all Π^0_1 classes can occur in this way. We describe the connection between the complexity of a basis for a group and an order for the group, and completely characterize the degree spectra of the set of bases for a group. We describe some restrictions on the possible degree spectra of the space of orders, including a connection to algorithmic randomness.

Computability Theory and Ordered Groups

Caleb J. Martin

M.S. Mathematics, University of Connecticut, Storrs, CT, 2010

B.S. Mathematics and Cognitive Science, Dartmouth College, Hanover, NH, 2009

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2015

APPROVAL PAGE

Doctor of Philosophy Dissertation

Computability Theory and Ordered Groups

Presented by

Caleb J. Martin, B.S., M.S.

Major Advisor ______________________
David Reed Solomon

Associate Advisor ______________________
Damir Dzhafarov

Associate Advisor ______________________
Stephen Flood

University of Connecticut

2015

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Reed Solomon, for all of his help and guidance. I am very grateful to have him as a mentor, and could not have done this without his dedication, patience, and insight. I would like to thank my parents, Linda and David Martin, for their constant love and support, for showing me what's important in life, and for all that they've done to help me become who I am today. Of course, I would also like to thank my wife, Rose, for always believing in me, for providing me with so much love and happiness (and food) over the last several years, and for generally being amazing. Finally, I would like to thank the rest of my family, friends, pets, colleagues, and teachers, especially Tom Morey, who first showed me the wonders mathematics has to offer.

Contents

Chapter 1

Introduction

1.1 Computability theory and effective algebra

In effective algebra, we examine familiar mathematical structures, e.g. groups, rings, vector spaces, and so on through the lens of computability. A structure in a finite language is said to be computable if its domain is a computable set and all of its functions and relations are computable. One of the main lines of inquiry when taking this approach is to ask to what extent the complexity of a given algebraic structure is predetermined by its internal structure, as opposed to the complexity of its elements or how its elements are arranged. For example, given a particular structure, one might ask if there is any difference in its isomorphism classes when restricted to computable isomorphisms, or which relations on the structure are computable, or which Turing degrees contain isomorphic copies of the structure, and so on.

A group is computable if determining membership in the group and evaluating the group operation are computable functions. Given a suitable coding into $\mathbb{N}$, familiar structures like the integers under addition, the nonzero rationals under multiplication, and a finite field under arithmetic are all examples of computable groups. However, the particulars of precisely how group elements are represented by the natural numbers (i.e. the presentation of the group) can dramatically affect whether or not the resulting object is actually computable: when we say something like "$\mathbb{Z}$ is computable" we really mean "there is a computable presentation of $\mathbb{Z}$".

Even if a group is computable, which in some sense means it is uncomplicated, it may have additional structure which is not computable. For example, it may or may not be possible to compute a basis for the group, an ordering on the group, the commutator subgroup, whether or not it is the Galois group of some extension of $\mathbb{Q}$, and so on. This work attempts to clarify exactly how complicated a basis and an order for a computable group might be.

1.2 Computable ordered groups

An order on a group $(G, \cdot)$ is a linear order $\leq$ (i.e. a total, transitive, antisymmetric binary relation) on its elements such that for all $a, b, c \in G$, $a \leq b$ implies $ac \leq bc$ and $ca \leq cb$. If the group has any torsion elements, there is no order on G that respects the group operation, since any element $a > 0$ with order n will satisfy $0 < a < na = 0$ or $0 > a > na = 0$. However, G being torsion-free is not a sufficient condition for the existence of an ordering. For example, suppose $G = \langle a, b \mid aba^{-1} = b^{-1} \rangle$. If $b > 1_G$, then $ab > a$, so $aba^{-1} > aa^{-1}$, so $b^{-1} > 1_G$ also, which is impossible. Similar reasoning applies if $b < 1_G$, so this group does not admit an order but is clearly torsion-free. However, this cannot occur if G is abelian, as shown by the following theorem.

Theorem 1.2.1 (Levi, 1942)**.** *Let G be an abelian group. Then G admits an order if and only if G is torsion-free.*

Henceforth, we restrict our attention to torsion-free abelian groups. Given a computable torsion-free abelian group, we can ask what can be said about the possible orders of that group. Computable ordered fields have been extensively studied, but computable ordered groups are less well-behaved. A natural first question is whether there must be a computable order, and in both cases this is false:

Theorem 1.2.2.

1. *(Rabin, 1960) There is a computable field which admits an order but not a computable order.*

2. *(Downey & Kurtz, 1986) There is a computable group which admits an order but not a computable order.*

To effectively analyze orders on a group G, it is more convenient to study the set of non-negative elements under each order. A positive cone of G is a subset P of G satisfying

1. For all $a \in G$, either $a \in P$ or $-a \in P$

2. If $a \in P$ and $-a \in P$, then $a = 0$

3. For all $a, b \in P$, $a + b \in G$

If we define $a \leq b$ to mean $b - a \in P$, it is easy to check that $\leq$ is an order on G. Similarly, if $P = \{x \in G : 0_G \leq_G x\}$, it is easy to check that P is the positive cone of $\leq_G$. If G is computable then the conditions above are Π^0_1, so the collection of all orders of some fixed G forms a Π^0_1 class.

Recall that a Π^0_1 class maybe also be defined as the set of infinite paths through a computable binary branching tree. A binary-branching tree is a set $T \subseteq 2^{<\omega}$ which is closed under initial segments, and a path through T is a function $f : \omega \to \{0, 1\}$ such that the sequence $\langle f(0), f(1), \ldots, f(n)\rangle$ is in T for all n. As a result, Π^0_1 classes are ubiquitous in mathematics, since many constructions can be viewed as following a binary-branching decision tree, guessing at an outcome at each node and terminating all branches extending nodes that correspond with invalid guesses.

To represent the collection of all orders of a computable group G in this way, enumerate the group elements as $G = \{0_G, g_0, g_1, g_2, \ldots\}$, and let $T_0 = \{\lambda\}$. Intuitively, we want group elements to correspond with levels of a tree, where a node on level n branches left if the corresponding group element is negative, and branches right if the corresponding group element is positive. At stage $s > 0$, we consider the possible extensions of each $\sigma \in T_{s-1}$ with $|\sigma| = s - 1$. Let $\tau \in \{\sigma 0, \sigma 1\}$. We say τ is a valid extension of σ if and only if

1. for all $i, j, k < s$, if $g_i + g_j = g_k$ and $\tau(i) = \tau(j)$, then $\tau(k) = \tau(i) = \tau(j)$
2. for all $i, j < s$, if $g_i = -g_j$ then $\tau(i) = 1 - \tau(j)$

Now we extend the tree by defining

$$T_s = \{\tau : |\tau| = s \wedge (\exists \sigma \in T_{s-1})(\tau \text{ is a valid extension of } \sigma)\}$$

and let $T = \bigcup_s T_s$. An infinite path through T is a choice of sign for each $g \in G$ such that sums of like-signed elements are given the same sign, and negations of elements are given the opposite sign. That is, the elements of $[T]$ are exactly the positive cones on G.

Obviously not all paths through this infinite binary tree will correspond with valid orders, such as the paths $\langle 000\ldots\rangle$ and $\langle 111\ldots\rangle$, but one can show that unless G is isomorphic to a subgroup of the rationals, infinitely many paths will. Given this correspondence, we will refer to positive cones of G and orders of G interchangeably, and we define $\mathbb{X}(G)$ to be the Π^0_1 class of orders on a computable (torsion-free abelian) group G.

1.3 Π^0_1 classes and representations

A major motivation for this work is the analogous situation with ordered fields. A computable field F is orderable if F is formally real, i.e. 1_F is not a sum of squares in F. Orders and positive cones of orders on fields are defined in exactly the same way as for abelian groups above, again with $\mathbb{X}(F)$ the Π^0_1 class of positive cones of F. In this case, the correspondence between spaces of orders and Π^0_1 classes is as exact as possible, in the following sense:

Theorem 1.3.1 (Metakides & Nerode, 1979)**.** *1. Let F be an orderable computable field. There is a Π^0_1 class $\mathcal{C}$ and a Turing degree preserving bijection from $\mathcal{C}$ to $\mathbb{X}(F)$.*

2. Let $\mathcal{C}$ be a Π^0_1 class. There is an orderable computable field F and a Turing degree preserving bijection from to $\mathbb{X}(F)$ to $\mathcal{C}$.

Of course, this means that any result about Π^0_1 classes immediately implies the corresponding result about orders on computable fields. For example, the low basis theorem states that every Π^0_1 class has a member of low degree, so every orderable computable field has an order of low degree. The same is true of the hyperimmune-free degrees. However, applying basis and anti-basis results only makes use of (1). The converse is a more useful statement, since it allows the construction of exotic Π^0_1 classes to prove the existence of orderable computable fields with the same property. For example, (Jockusch & Soare, 1972) constructed a Π^0_1 class such that any two distinct elements have incomparable Turing degree. It follows from (2) that there is an orderable computable field such that any two distinct orders are Turing incomparable.

For orderable computable groups, we have already seen that the analog of (1) holds, but (2) fails. An immediate obstacle is that if $\leq$ is an order on a computable group G, so is $\leq^*$, where $a \leq^* b$ if and only if $b \leq a$. However, it is clear that $\leq$ and $\leq^*$ are Turing equivalent, so it is impossible for the Π^0_1 class constructed by Jockusch and Soare to represent the space of orders on a computable group. As a result, we will shift our attention to comparing the degree spectrum of the collection of orders for G, $\mathrm{Spec}(\mathbb{X}(G)) = \{\deg(P) : P \in \mathbb{X}(G)\}$, to the degree spectrum of a Π^0_1 class.

As effectively closed subsets of $2^{<\omega}$, Π^0_1 classes have been extensively studied and are fairly ubiquitous in mathematics. For example, given a theory T of first order logic, the set of complete consistent extensions of T forms a Π^0_1 class. This is easy to see, since in the above construction which included extensions of a node with length n by 0 or 1 depending on the consistency of assigning a sign to the n-th group element, one can replace an enumeration of group elements with an enumeration of sentences and replace sign with satisfaction. Similarly, the Stone space $S(B)$ of a Boolean algebra B is the set of ultrafilters on B, which forms a Π^0_1 class. Unsurprisingly, combinatorics is rife with examples of structures which form Π^0_1 classes, such as the collection of k-colorings of a computable graph, the set of Hamiltonian or Eulerian paths on a highly recursive graph, and many others. In fact, (Cenzer and Remmel,

1998) showed that both of the previous two examples can present the Π^0_1 class of separating sets for any pair of disjoint c.e. sets.

1.4 Linear independence and rank

A group is divisible if for all $g \in G$ and all positive integers n, there is some $d \in G$ with $g = nd$. A group D is a divisible closure of G if D is divisible there is an embedding of $G \hookrightarrow D$ where each nonzero $d \in D$ is such that nd is the image of g for some $g \in G$ and $n \in \mathbb{N}$. Smith (1981) showed that if G is a computable abelian group, there is a computable group D such that D is a divisible closure of G and there is a computable embedding that demonstrates this. Solomon (2003) showed that if D is a divisible closure of G, then G is orderable if and only if D is orderable, and every order on G extends uniquely to an order on D. Furthermore, there is a Turing degree preserving bijection from $\mathbb{X}(G)$ to $\mathbb{X}(D)$. This means that when investigating the degree spectrum of $\mathbb{X}(G)$, we may assume that G is divisible whenever convenient.

If G is a torsion free abelian group, we say elements $g_0, \ldots, g_n$ are linearly independent if for all integers $\alpha_0, \ldots, \alpha_n$, $\alpha_0 g_0 + \cdots + \alpha_n g_n = 0_G$ if and only if $\alpha_i = 0$ for each i. An infinite subset of G is linearly independent if all of its finite subsets are linearly independent. As usual, a maximal set of linearly independent elements of G is called a basis, and the cardinality of any basis for G is called the rank of G. The rank of G plays an important role in the possible degree spectra of G, as follows.

Theorem 1.4.1 (Solomon, 2003). *Let G be a computable torsion free abelian group.*

1. *If G has rank 1, then G has precisely two orders, both of which are computable.*

2. *If G has finite rank $n > 1$, then G has orders of all Turing degrees.*

3. If G has infinite rank, then G has orders of all Turing degrees above $\mathbf{0}'$.

The case when G has rank 1 is clear, since the divisible closure of any rank 1 group is isomorphic to $\mathbb{Q}$. Since $\mathbb{Q}$ has only the usual order $\leq_{\mathbb{Q}}$ and its reverse $\leq_{\mathbb{Q}}^*$, both of which are computable, the same is true of G.

In the case when G has finite rank at least two, the idea is to assume G is divisible, fix independent elements $\{a, b\}$, fix an infinite set D of Turing degree $\mathbf{d}$, and define a map from G to $\mathbb{R}$ by $a \mapsto 1$ and $b \mapsto \sum_i 2^{-i}\chi_D(i)$. Noting that the image of b is an irrational number with Turing degree $\mathbf{d}$ and extending this map linearly induces an order with degree $\mathbf{d}$ on the subgroup H of G generated by $\{a, b\}$, and combining this lexicographically with a computable order on the subgroup $G \setminus H$ yields an order of G with degree $\mathbf{d}$.

In the case when G has infinite rank, the proof is the same, but while a finite basis can be searched for and found computably, computing an infinite basis requires $\mathbf{0}'$. As a result, the argument works if $\mathbf{d} \geq_T \mathbf{0}'$. We will refine this result shortly.

1.5 Overview

Since the characterization of $\mathbb{X}(G)$ based on the rank of G given above makes essential use of a basis for G, before examining $\mathbb{X}(G)$ any further directly, we explore the connection between orders on G and bases for G.

Chapter 2 is devoted to analyzing the possible degree spectrum of the set of bases for G, which actually admits a very simple characterization. In particular, we will see that the Turing degrees of possible bases for G are exactly the cone above some c.e. degree, and that $\mathrm{Spec}(\mathbb{X}(G))$ contains these degrees.

In Chapter 3, we construct an example of a group with orders in every degree, but no basis below $\mathbf{0}'$. (Kach, Lang, & Solomon, 2013) constructed a group with computable orders but no noncomputable orders below some c.e. set, which shows that $\mathrm{Spec}(\mathbb{X}(G))$ need not be upwards closed. We will strengthen this by constructing a group with a computable order where $\mathrm{Spec}(\mathbb{X}(G))$ avoids the cone below some high degree.

In addition to containing the cone above $\mathbf{0}'$, $\mathrm{Spec}(\mathbb{X}(G))$ must contain many low degrees. This was originally shown by Kach, Lang, Solomon, & Turetsky by applying the low basis theorem to find a low path, finding a subtree with no paths computable from any of the low paths found so far, applying the low basis theorem to this subtree, and continuing inductively. The same argument applies for any class of degrees which form a basis for Π^0_1 classes and are fixed by relativizing to a member of the class, such as the hyperimmune-free degrees. Note that any Π^0_1 class with no computable element has these properties as well, but this does not hold of Π^0_1 classes in general. In fact, there are Π^0_1 classes containing only computable elements and elements with degree above $\mathbf{0}'$. However, $\mathrm{Spec}(\mathbb{X}(G)$ has infinitely many low degrees and infinitely many hyperimmune-free degrees, even if it has computable members.

In Chapter 4 we will show a more general result that implies this fact. We will also show in Chapter 4 that $\mathbb{X}(G)$ contains no 1-random elements, and that it cannot be a thin Π^0_1 class.

Chapter 2

Orders and bases

2.1 The degrees of bases

Let G be a computable torsion-free abelian group with infinite rank, and consider the collection $\mathcal{B}_G$ of bases for G. A natural question is to investigate how complicated a basis for G might be, in the sense of its degree spectrum,

$$\mathrm{Spec}(\mathcal{B}_G) = \{\mathbf{d} : \mathbf{d} = \deg B \text{ for some } B \in \mathcal{B}_G\}.$$

In general, the degree spectra of particular objects may not admit a simple description, but $\mathrm{Spec}(\mathcal{B}_G)$ may be completely characterized as follows. While there are many bases of G, some of which may have arbitrary (extraneous) information coded into them, each one tells us essentially the same thing about G: when elements of G are algebraically independent. With this in mind, define

$$I_G = \{\langle g_1, g_2, \dots, g_n \rangle : \{g_1, \dots, g_n\} \text{ is algebraically independent in } G\}.$$

If B is a basis for G, then I_G should be B-computable. In fact, $\mathbf{i}_G = \deg I_G$ exactly characterizes the information content (i.e. Turing degrees) of bases of G:

Theorem 2.1.1. *Let G be a computable torsion-free abelian group with infinite rank, and let $\mathrm{Spec}(\mathcal{B}_G)$ and $\mathbf{i}_G$ be as above. Then $\mathrm{Spec}(\mathcal{B}_G) = \{\mathbf{d} : \mathbf{d} \geq \mathbf{i}_G\}$.*

Proof. Fix some effective enumeration of G as $\{0_G, g_1, g_2, \dots\}$ with $g_0 = 0_G$. First, we show that I_G can compute a canonical basis $\hat{B}$ of G, and conversely, any $B \in \mathcal{B}_G$ can compute I_G. This implies that $\deg \hat{B} \leq \mathbf{i}_G \leq \deg \hat{B}$, which shows $\mathbf{i}_G = \deg \hat{B} \in \mathrm{Spec}(\mathcal{B}_G)$, and that the degree of any basis lies above $\mathbf{i}_G$.

Using I_G and the enumeration of G above, compute a canonical basis $\hat{B}$ of G as follows. Let $B_0 = \{g_1\}$, and for each $s > 1$, let B_s be $B_{s-1} \cup \{g_s\}$ if $B_{s-1} \cup \{g_s\}$ is independent (i.e. in I_G), otherwise keep $B_s = B_{s-1}$.

We claim $\hat{B} = \bigcup_s B_s$ is a basis for G. Clearly $\hat{B}$ is a basis for G, since $\hat{B}$ is an independent set by construction, and every element g_n of G is either added to $\hat{B}$ at stage n in the construction above or is found to be algebraically dependent on $B_{n-1} \subset \hat{B}$ at stage n, so $\mathrm{span}(\hat{B}) = G$.

We also have that $\hat{B}$ is computable from I_G, since to check whether or not some $g_n \in \hat{B}$, run the construction of the B_s above until B_n is defined. Using I_G as an oracle, we either put g_n into B_n, in which case $g_n \in \hat{B}$; or we keep it out, and g_n is never considered for membership in any future B_m with $m > n$, in which case $g_n \notin \hat{B}$.

On the other hand, let $B = \{b_i : i \in \omega\} \in \mathcal{B}_G$ be any basis. B can compute whether or not $\langle g_{i_1}, \dots, g_{i_n} \rangle$ is in I_G by first expressing each g_{i_j} as a finite linear combination of the basis elements in B and then using Gaussian elimination to determine if these elements are independent. Both of these operations can be done computably.

This establishes that $\mathrm{Spec}(\mathcal{B}_G)$ contains the cone above $\mathbf{i}_G$, so to prove the theorem, it remains to show that any Turing degree $\mathbf{d} \geq \mathbf{i}_G$ is the degree of some basis $B_D \in \mathcal{B}_G$. Let D be a set of degree $\mathbf{d}$. Since $\mathbf{d} \geq \mathbf{i}_G$, D can compute I_G, so D can compute $\hat{B}$. List the elements of $\hat{B}$ as $\{\hat{b}_0, \hat{b}_1, \dots\}$, written in the order they enter in the construction of $\hat{B}$, which is the same order they appear in our original enumeration of G. Define B_D to be the set whose n^{th} element is $\hat{b}_n$ if $n \in D$ and $2\hat{b}_n$ if $n \notin D$.

That B_D is a basis is clear, since it is equal to the basis B modulo taking scalar multiples of each element, which has no effect on linear independence. It is also clear that $B_D \leq_T D$. But since B_D is a basis, it can compute $\hat{B}$. In particular, for any n it can compute the n^{th} element $\hat{b}_n$ of $\hat{B}$, and then $n \in D$ if and only if that $\hat{b}_n \in B_D$, since $\hat{b}_n$ will either be the n^{th} element of B_D, or is algebraically dependent on the n^{th} element of B_D, according to whether or not $n \in D$. That is, $D \leq_T B_D$, so $\deg B_D = \mathbf{d}$, as desired, so all $\mathbf{d} \geq \mathbf{i}_G$ are in $\mathrm{Spec}(\mathcal{B}_G)$. □

2.2 The degree of the independence relation

To fully understand $\mathrm{Spec}(\mathcal{B}_G)$, the only question that remains is where $\mathbf{i}_G$ might sit in the Turing degrees. Since $\mathbf{0}'$ can search for algebraic dependencies, $\mathbf{i}_G \leq \mathbf{0}'$. Looking more closely, if G is computable then I_G is a Π^0_1 set, since an n-tuple $(g_1, \dots, g_n)$ is a member of I_G if and only if for all nonzero linear polynomials $f \in \mathbb{Z}[x_1, \dots, x_n]$, $f(g_1, \dots, g_n) \neq 0$, and this latter condition is computable. So $\mathbf{i}_G = \deg I_G = \deg \overline{I_G}$ is a c.e. degree, and in fact, this is the only restriction on $\mathbf{i}_G$.

Theorem 2.2.1. *For any c.e. degree* $\mathbf{c}$*, there is a computable torsion-free abelian* G *of infinite rank with* $\deg I_G = \mathbf{c}$*.*

Proof. Fix some c.e. C with degree $\mathbf{c}$, and let f be a 1-1 computable function with $\operatorname{ran} f = C$, so C is enumerated as $\{f(0), f(1), \dots\}$. We build G classically isomorphic to $\bigoplus_\omega \mathbb{Z}$ as follows.

Fix some coinfinite set $S = \{a_1, b_1, a_2, b_2, \dots\} \subset \omega$. The group G will be a subgroup of the $\mathbb{Z}$-linear combinations of S, where if $f(s) = j$, then $b_j = (2s+1)a_j$. That is, we define G to be the group whose elements are formal sums

$$G = \left\{ \sum_{i \in I} \alpha_i a_i + \sum_{j \in J} \beta_j b_j : \alpha_i, \beta_j \in \mathbb{Z} \setminus \{0\} \wedge (\forall i \in I)(\forall s < |\alpha_i|)(f(s) \neq j) \wedge I, J \text{ finite} \right\}$$

Addition on G is defined componentwise with a reduction procedure to maintain the restrictions on the a_i coefficients. Specifically, to add $g_1 = \sum_{i \in I_1} \alpha_i a_i + \sum_{j \in J_1 \beta_j b_j}$ and $g_2 = \sum_{i \in I_2} \gamma_i a_i + \sum_{j \in J_2} \delta_j b_j$, first pad each sum using terms with zero coefficients until the index sets agree. That is, write $g_1 = \sum_{i \in I_1 \cup I_2} \alpha_i a_i + \sum_{j \in J_1 \cup J_2} \beta_j b_j$, where $\alpha_i = 0$ for $i \in I_2 \setminus I_2$ and $\beta_j = 0$ for $j \in J_2 \setminus J_1$, and similarly, write $g_2 = \sum_{i \in I_1 \cup I_2} \gamma_i a_i + \sum_{j \in J_1 \cup J_2} \delta_j b_j$. Adding these sums componentwise gives $\sum_{i \in I_1 \cup I_2} (\alpha_i + \gamma_i) a_i + \sum_{j \in J_1 \cup J_2} (\beta_j + \delta_j) b_j$.

Omit any terms with coefficient zero, giving $\sum_{k \in K} \eta_k a_k + \sum_{l \in L} \zeta_l b_l$. Finally, we check whether any of the ζ_k are too large, and if so, we use an appropriate group relation to reduce them. That is, for each $k \in K$ we check if $f(s) = k$ for some $s < |\eta_k|$. If not, we leave $\eta_k a_k$ in our sum. If there is such an s (note that there can be at most one such s), assume $\eta_k > 0$, and we apply the relation $b_k = (2s+1)a_k$ to write

$$\eta_k a_k = \eta_k a_k - (2s+1)a_k + b_k = (\eta_k - 2s - 1)a_k + b_k$$

. We claim that the new coefficient $c_k = \eta_k - (2s+1)$ satisfies $|c_k| \leq s$, hence is a valid coefficient for a_k in a formal sum in G. To show this, we know $\eta_k = \alpha_k + \gamma_k$ and $|\alpha_k| \leq s$, $|\gamma_k| \leq s$, and $s < |\eta_k| = |\alpha_k + \delta_k|$. Therefore, $0 < \alpha_k \leq s$ and $0 < \gamma_k \leq s$. It follows that $s < \alpha_k + \gamma_k \leq 2s$, so $s + 1 \leq \alpha_k + \gamma_k \leq 2s$. Then $-s \leq \alpha_k + \gamma_k - 2s - 1 \leq -1$, which implies $|\alpha_k + \gamma_k - 2s - 1| \leq s$, as required. If $\eta_k < 0$, we write $\eta_k a_k = (\eta_k + 2s + 1)a_k - b_k$, and similar reasoning applies. So if $f(s) = k$ for some $s \leq |\eta_k|$, we reduce the sum for g by replacing any terms of the form $\eta_k a_k$ by $(\eta_k - 2s - 1)a_k$ if $\eta_k > 0$, replacing $\zeta_k b_k$ by $(\zeta_k + 1)b_k$ if $k \in K$ and introducing $1b_k$ into the formal sum if $k \notin K$.

This is a finite process and f is computable, so addition can be done computably and G is a computable group. Also, it is clear that G is torsion-free and has infinite rank. G is closely related to the free group generated by S, but we have made b_j dependent on a_j if and only if j is eventually enumerated into C. The reduction process described above is the only relation on the elements of G, so a presentation for G is

$$G = \langle S \mid \text{for each } s,\ b_{f(s)} = (2s+1)a_{f(s)} \rangle.$$

That is, a_i and a_j are dependent if and only if $i = j$, b_i and b_j are dependent if and only if $i = j$, and a_i and b_j are dependent if and only if $i = j \in C$. Therefore, a basis for G is $\{a_i, b_j : i \in \omega, j \in \overline{C}\}$.

We need to show that $I_G \leq_T C$ and vice versa. As noted above, C computes the basis $\{a_i, b_j : i \in \omega, j \in \overline{C}\}$ for G, and since any basis can compute I_G, we have $I_G \leq_T C$. The reverse is similarly straightforward: given knowledge of I_G, we can compute that $n \in C$ if and only if $\langle a_n, b_n \rangle \notin I_G$, so $C \leq_T I_G$ and the result follows. $\square$

Chapter 3

Separating results

3.1 Disconnecting bases from orders

Again, let G be a computable torsion-free abelian group with infinite rank, and let $\mathcal{B}_G$ be the collection of bases for G. Any such G must have a $0'$-computable basis, since the independence relation on G is computably enumerable (and we have seen that the Turing degrees of bases of G are precisely the cone above the degree of the independence relation). Furthermore, if B is a basis, G has orders in every Turing degree above $\deg B$.

Two useful facts follow immediately from this: every such G has orders of all degrees above $0'$, and if G has a computable basis, G has orders of every degree. However, a computable basis is not necessary for the presence of orders of all degrees. In fact, it is possible that the only connection between bases and orders be the implied presence of orders of all degrees in the cone above $\mathbf{0}'$.

Theorem 3.1.1. *There is a computable torsion-free abelian group G with infinite rank such that G has no basis with degree strictly below $\mathbf{0}'$ but has orders in every Turing degree.*

Proof. We will build $G \cong \mathbb{Q}^\omega \oplus H$, where H has a computable order but every basis of H computes $\mathbf{0}'$. Since $\mathbb{Q}^\omega$ has a presentation with a computable basis, hence has an order $\leq_{\mathbf{d}}$ in every degree $\mathbf{d}$, combining this lexicographically with the computable order on H produces an order of G with degree $\mathbf{d}$. Hence we need only ensure that every basis of H computes $0'$.

The group $(H,+_H)$ will be constructed in stages via a finite injury construction with H_s denoting the finite set of elements in H at stage s. We define a partial function $+_s$ on H_s to specify the addition facts declared at the end of stage s. To ensure that H is computable, we will have $H_s \subseteq H_{s+1}$ for all s and $H = \bigcup_s H_s$. We need to maintain that if $x +_s y = z$, then $x +_t y = z$ for all $t \geq s$. Finally, we must ensure that if $x, y \in H_s$, there is some $t \geq s$ and some $\hat{x}, z \in H_t$ with $x +_t y = z$ and $x + \hat{x} = 0_H$. In the end, H will be classically isomorphic to $\mathbb{Q}^\omega$.

To define the the addition on H, we use an approximation $\{b_0^s, b_1^s, \ldots, b_s^s\} \subseteq H_s$ to an initial segment of a basis for H. During the construction, each approximate basis element will be changed at most once, i.e. $b_i^s \neq b_i^{s+1}$ for at most one s for any given i. This ensures that $b_i = \lim_s b_i^s$ exists for all i, and the set $\{b_0, b_1, \ldots\}$ will be a basis for H. The basis element b_0^s will play a special role, and be equal for all s.

To ensure that every basis of H computes $0'$, we will encode $0'$ directly into the independence relation of H. In particular, at stage e we will define the basis element b_e^e, and for all $s \geq e$, if $\Phi_e^s(e)\uparrow$ we will maintain $b_e^s = b_e^e$ so $\{b_0^s, b_e^s\}$ is an independent set. However, if $\Phi_e^s(e)\downarrow$ and e enters $0'$, we will set $b_e^s = q b_0^{s+1}$ for some rational number

q, making $\{b_0^{s+1}, b_e^s\}$ linearly dependent. We will then redefine b_e^{s+1} to be a new group element and keep this fixed as a basis element for the rest of the construction.

To recover $0'$ from the independence relation of H, we need to use the independence relation of H to compute whether $e \in 0'$ for each $e \geq 1$. To do so, go to stage e of the construction and find the element b_e^e. If $e \in 0'$ by stage e, then of course $e \in 0'$ and we're done. Otherwise, use the independence relation to determine whether or not $\{b_0, b_e^e\}$ is dependent. If so, then $e \in 0'$, and if not, then $e \notin 0'$.

In order to define the group elements and operation, each element $g \in H_s$ will be assigned a $\mathbb{Q}$-linear sum over the approximate basis at stage s, i.e. $q_0^s b_0^s + \cdots + q_n^n b_n^s$ where $n \leq s$, $q_i \in \mathbb{Q}$, and $q_n^s \neq 0$. This assignment will be injective, with 0_h assigned the empty sum. Informally, we will write that g is equal to this sum. Then the partial addition function $+_s$ on H_s will be defined by $x +_s y = z$ if and only if the corresponding sums for x and y add up to the sum for z, discarding trailing zeros. This operation is clearly associative and commutative, and automatically satisfies $x +_s 0_H = x$ for all $x \in H_s$. Furthermore, whenever we introduce dependencies involving b_e^s in order to code $0'$ into the independence relation, we will redefine the sums assigned to all elements with nonzero q_e term to maintain all previously specified addition facts.

Construction: Fix a computable coding of $\mathbb{Q}^{<\omega}$. We assume that at stage s, at most one number enters $0'$. At stage 0, set $H_0 = \{0, 1\}$. We assign 0 the empty sum, so $0_H = 0$, and assign 1 the sum $1b_0^0$. Define $+_0$ to be the relation with $0_H +_0 0_H = 0_H$ and $0_H +_0 b_0^0 = b_0^0 +_0 0_H = b_0^0$. Define the approximate basis at stage 0 to be $\{b_0^0\}$.

In general, at the end of stage s we will have a finite set H_s and an approximate basis $\{b_0^s, \ldots, b_s^s\}$, and each $h \in H_s$ will be assigned a sum $h = q_0^s b_0^s + \ldots q_n^s b_n^s$ as above. Equivalently, we can set $h = q_0^s b_0^s + \ldots q_s^s b_s^s$ by padding the sum with trailing zeros.

The partial function $+_s$ is defined on $(x,y) \in H_s^2$ if any only if there is a $z \in H_s$ such that the sums assigned to x and y add up componentwise to the sum assigned to z. For later stages $s+1$, we have two cases to consider.

Case 1: Suppose no number $e \leq s$ enters $0'$ at stage $s+1$. In this case, we do nothing with regards to the coding requirements. Define $b_e^{s+1} = b_e^s$ for all $e \leq s$. Put each element of H_s into H_{s+1}, and assign it the same sum. That is, for each $g = \sum_{i \leq s} q_i^s b_i^s$ in H_s, put $\sum_{i \leq s+1} q_i^{s+1} b_i^{s+1}$ into H_s with $q_i^{s+1} = q_i^s$ for each $i \leq s$ and $q_{s+1}^{s+1} = 0$. This ensures that $x +_s y = z$ implies $x +_{s+1} y = z$ for all $x, y, z \in H_s$. Additionally, we add two new elements to H_{s+1}. The first is assigned the sum $1b_{s+1}^{s+1}$, and the second is assigned the sum $q_0^{s+1} b_0^{s+1} + \cdots + q_n^{s+1} b_n^{s+1}$, where $\langle q_0^{s+1}, \ldots, q_n^{s+1} \rangle$ is the least element in our fixed coding of $\mathbb{Q}^{<\omega}$ not yet assigned to an element of H_{s+1} with $n \leq s$, $q_n^{s+1} \neq 0$.

Case 2: Now suppose some $e \leq s$ enters $0'$ at stage $s+1$, in which case we must act to satisfy the coding requirements. By assumption, there is exactly one such e. For each $i \leq s$ with $i \neq e$, set $b_i^{s+1} = b_i^s$. We need to choose a suitable rational number q. For now, ensure that q is such that the code for $\langle q \rangle$ is larger than any number used in the construction up to stage s. We will also put an additional restriction on q to ensure that H has a computable order, but those do not otherwise affect the construction and will be specified later. Assign b_e^s to the sum qb_0^{s+1}, so $\{b_0^{s+1}, b_e^s\}$ is a dependent set. More generally, for each $g \in H_s$ we assign g the same sum as an element of H_{s+1}, but replace each instance of b_i^s with b_i^{s+1} for $i \neq e$, and replace each instance of b_e^s with qb_0^{s+1}. That is, if $g = q_0^s b_0^s + \cdots + q_e^s b_s^s + \cdots + q_n^s b_n^s$ in H_s, the corresponding sum assigned to g as an element of H_{s+1} is $g = (q_0^s + qq_e^s) b_0^{s+1} + \cdots + 0b_e^{s+1} + \cdots + q_n^s b_n^{s+1}$. That is, set $q_e^{s+1} = 0$, set $q_0^{s+1} = q_0^s + qq_e^s$, and set $q_i^{s+1} = q_i^s$ for all $i \neq 0, e$. Furthermore, we assume that q is chosen so this assignment of sums remains injective.

Additionally, we add three new elements to H_{s+1}. As in the previous case, we add a new approximate basis element assigned the sum $1b_{s+1}^{s+1}$, and we also add an element assigned the sum $1b_e^{s+1}$. Finally, we add an element assigned the sum $q_0^{s+1}b_0^{s+1} + \cdots + q_n^{s+1}b_n^{s+1}$, where $\langle q_0^{s+1}, \ldots, q_n^{s+1}\rangle$ is the least element in our fixed coding of $\mathbb{Q}^{<\omega}$ not yet assigned to an element of H_{s+1} with $n \leq s$, $q_n^{s+1} \neq 0$.

At this point, we need to pause and verify that several conditions are satisfied for the overall construction to work. First, we must ensure that each of the approximate formal sums converge.

Lemma 3.1.2. *For each $g \in H$, there is a stage t such that g is assigned a sum $g = q_0^t b_0^t + \cdots + q_n^t b_n^t$ that is not changed. That is, at all stages $u \geq t$, g is assigned the sum $q_0^u b_0^u + \cdots + q_n^u b_n^u$ with $q_i^u = q_i^t$ and $b_i^u = b_i^t$ for all $i \leq n$.*

Proof. When g first enters H at some stage s, it is assigned a sum $g = q_0^s b_0^s + \cdots + q_n^s b_n^s$. The only part of the construction that can cause this sum to change is when we act to satisfy a coding requirement, say at stage t. In this case, some basis element b_e^t is made dependent via $b_e^t = qb_0^{t+1}$, and a new basis element b_e^{t+1} is introduced, which will never change (or be part of the sum assigned to g). That is, each approximate basis element b_i involved in the sum assigned to g can be changed at most once, so the sum for g can be changed at most n times during the construction. □

We also need to verify that all finite tuples of rational numbers will correspond to a sum in H.

Lemma 3.1.3. *For each rational tuple $\langle q_0, q_1, \ldots, q_n\rangle$ with $q_n \neq 0$, there is a $g \in H$ such that the limiting sum for g is $q_0 b_0 + \cdots + q_n b_n$.*

Proof. Assume otherwise, and let $\langle q_0, \dots, q_n \rangle$ be the least such tuple which does not correspond to any element of H in the limit. Let s be a stage by which $b_0^s, \dots, b_n^s$ have reached their limits (again, since each b_i can be redefined at most once), and each tuple with code less than $\langle q_0, \dots, q_n \rangle$ has appeared as the limiting sum of an element of H. By construction, at stage $s+1$ either there is an element which is assigned the sum $q_0 b_0^{s+1} + \cdots + q_n b_n^{s+1}$, or we add a new element to H and assign it this sum. In either case, since the basis elements involved have reached their limits, the tuple in question is the limiting tuple of an element of H. □

We also need to verify that the operation $+_H$ has the required properties.

Lemma 3.1.4. *If $x +_s y = z$, then the limiting sums for x and y add to form the limiting sum for z.*

Proof. In both cases of the construction above, at stage $s+1$ we ensured that if $x +_s y = z$, then $x +_{s+1} y = z$. By induction $x +_t y = z$ for all $t \geq z$, and the result follows. □

Lemma 3.1.5. *For each pair $x, y \in H_s$, there is a $t \geq s$ and elements $\hat{x}, z \in H_t$ such that $x +_t y = z$ and $x +_t \hat{x} = 0_H$.*

Proof. Fix $x, y \in H_s$, and wait until a stage $s' \geq s$ at which x and y have been assigned their limiting sums. Let $q_0 b_0^{s'} + \cdots + q_n b_n^{s'}$ be the sum of these limiting sums. By the previous lemma, there is a stage $t_0 \geq s'$ such that some $z \in H_{t_0}$ is assigned the sum $q_0 b_0^{t_0} + \cdots + q_n b_n^{t_0}$, and therefore $x +_{t_0} y = z$. By the same reasoning, there is a stage $t_1 \geq s'$ and a $\hat{x} \in H_{t_1}$ which is assigned the sum $(-q_0) b_0^{t_1} + \cdots + (-q_n) b_n^{t_1}$, and therefore $x +_{t_1} \hat{x} = 0_H$. Taking $t = \max\{t_0, t_1\}$ gives the desired result. □

All that remains of the construction is to ensure that H has a computable order. As mentioned earlier, this will place additional restrictions on our choice of q in case when a number enters $0'$ at stage s, but will not otherwise interact with the details of the construction given above. At each stage s, we will define a binary relation $\leq_s$ on H_s as an approximation to an order on H. To make the order $\leq_H = \bigcup_s \leq_s$ computable, we will ensure that $x \leq_s y$ implies $x \leq_t y$ for all $t \geq s$. At any given stage, $\leq_s$ will not be defined on all pairs of elements of H_s, but we will ensure that for all $x, y \in H_s$, there is some $t \geq s$ such that $x \leq_t y$ or $y \leq_t x$.

To help define the order on H, we build a Δ^0_2 map f from H to $\mathbb{R}$. In particular, our map will send $b_0 = b^s_0$ to $1_\mathbb{R}$, and will send $b_i = \lim_s b^s_i$ to $\sqrt{p_i}$, where p_i is the i-th prime number. Note that the set $\{\sqrt{p_i} : i \geq 1\}$ is algebraically independent over $\mathbb{Q}$. We then extend the map linearly in the natural way: if $g \in H$ is assigned the limiting sum $\sum_{i \geq 0} q_i b_i$, then our map will send g to $q_0 + \sum_{i \geq 1} q_i \sqrt{p_i}$ in $\mathbb{R}$.

During the construction, we will approximate this map by maintaining rational intervals around the image of each basis element. That is, at stage s, for each $i \geq 1$ we assign each approximate basis element b^s_i in H_s a rational interval $(a^s_i, \hat{a}^s_i)$ with $a, \hat{a} \in \mathbb{Q}$, $a^s_i < \sqrt{p_i} < \hat{a}^s_i$, and $\hat{a}^s_i - a^s_i < 2^{-s}$. The element b^s_0 is always assigned $1_\mathbb{R}$. We will ensure that the interval $(a^s_i, \hat{a}^s_i)$ assigned to b^s_i is such that in the limiting order $\leq_H$, $a^s_i b_0 \leq_H b^s_i \leq \hat{a}^s_i b_0$, which places ordering constraints on b^s_i relative to b_0.

Since each $g \in H_s$ is given by some sum $g = q^s_0 b^s_0 + \cdots + q^s_n b^s_n$, the constraints on each b^s_i for $i \leq n$ place a corresponding interval constraint on g. The endpoints for this interval may be determined in a natural way from each of the intervals $(a^s_i, \hat{a}^s_i)$ with $i \leq n$. To define $\leq_s$ on H_s, we look at the interval constraints for each pair of distinct $x, y \in H_s$. If the intervals are disjoint in $\mathbb{R}$, we define $x \leq_s y$ if the interval for x lies to the left of the interval for y, and we define $y \leq_s x$ if the reverse is true.

If the intervals for x and y are not disjoint, we do not define $\leq_s$ on the pair (x, y).

Formally, at the end of stage s, we will have assigned rational intervals $(a_i^s, \hat{a}_i^s)$ to each b_i^s with $i \leq s$ with $a_i^s <_{\mathbb{R}} \sqrt{p}_i <_{\mathbb{R}} \hat{a}_i^s$ and $\hat{a}_i^s - a_i^s < 2^{-s}$. As before at stage $s+1$, we split into two cases, according to whether or not some $e \leq s$ enters $0'$ at stage $s+1$.

Case 1: Suppose no $e \leq s$ enters $0'$ at stage $s+1$. Then for each $i \leq s$, define a_i^{s+1} and $\hat{a}_i^{s+1}$ to be rational number such that $a_i^s < a_i^{s+1} < \sqrt{p_i} < \hat{a}_i^{s+1} < \hat{a}_i^s$ and $\hat{a}_i^{s+1} - a_i^{s+1} < 2^{-(s+1)}$. Since each of the interval constrains for the b_i are nested, the corresponding interval constraints for all other $g \in H_s$ are nested. It follows that $x <_s y$ implies $x <_{s+1} y$ for all $x, y \in H_s$. For the new approximate basis element b_{s+1}^{s+1}, we assign an interval $(a_{s+1}^{s+1}, \hat{a}_{s+1}^{s+1})$ with $a_{s+1}^{s+1} < \sqrt{p_{s+1}} < \hat{a}_{s+1}^{s+1}$ and $\hat{a}_{s+1}^{s+1} - a_{s+1}^{s+1} < 2^{-(s+1)}$. Furthermore, we find the interval constraint for the other new element assigned the least unused tuple of rationals, and determine its $\leq_{s+1}$-ordering relative to all existing elements of H_{s+1} using their interval constraints in the manner stated above.

Case 2: Now suppose some $e \leq s$ enters $0'$ at stage $s+1$. We will have introduced a new dependence relation by setting $b_e^s = q b_0^{s+1}$. Since we have specified an interval constraint $(a_e^s, \hat{a}_e^s)$ for b_e^s, choose q to lie in this interval. Since there are infinitely many such q, we are free to choose q with $\langle q \rangle$ arbitrarily large, as well as ensuring that assignment of sums to all elements of H_{s+1} remains injective. Additionally, since the interval constraint for the new sum for $b_e^s = q b_0^{s+1}$ is within the previous interval constraint, the new stage $s+1$ sum assignments will have interval constraints within the stage s interval constraints. This ensures that $x <_s y$ implies $x <_{s+1} y$ for all $x \neq y \in H_s$.

For the new elements added to H_{s+1}, we choose an interval $(a_{s+1}^{s+1}, \hat{a}_{s+1}^{s+1})$ for b_{s+1}^{s+1} such that $a_{s+1}^{s+1} < \sqrt{p_{s+1}} < \hat{a}_{s+1}^{s+1}$ and $\hat{a}_{s+1}^{s+1} - a_{s+1}^{s+1} < 2^{-(s+1)}$. Similarly, we choose an interval $(a_e^{s+1}, \hat{a}_e^{s+1})$ for b_e^{s+1} such that $a_e^{s+1} < \sqrt{p_e} < \hat{a}_e^{s+1}$ and $\hat{a}_e^{s+1} - a_e^{s+1} < 2^{-(s+1)}$. Using these interval constraints for all the b_i^{s+1}, we compute interval constrains for all other elements based on their sum.

To complete the proof, we need to verify that the relation $\leq_H$ constructed above is actually an order on H with the required properties.

Lemma 3.1.6. *For all $x, y \in H_s$, if $x \leq_s y$, then $x \leq_t y$ for all $t \geq s$.*

Proof. From the definition of $\leq_s$, for each $x \in G_s$ the constraining interval for x at stage $s+1$ is contained in the constraining interval for x at stage s. The same is true of y, so if $x \leq_s y$ then $x \leq_{s+1} y$. By induction, we have this for all $t \geq s$. □

Lemma 3.1.7. *For each $x, y \in H_s$, there is a stage $t \geq s$ such that $x \leq_t y$ or $y \leq_t x$.*

Proof. Since $x \leq_s x$ for all $x \in H_s$, we assume x, y are distinct elements of H_s. Let $s' \geq s$ be a stage at which x and y have reached their limiting sums. At each stage $t \geq s'$, the interval constraints around x and y are nested and converge (in $\mathbb{R}$) to the value of their limiting sums, with each b_t^t replaced by $\sqrt{p_i}$. Because the real numbers $\{\sqrt{p_j} : j \geq 1\}$ are algebraically independent over $\mathbb{Q}$ and the limiting sums of x and y are distinct, there is a stage $t \geq s'$ at which these interval constraints become disjoint. It t is the first such stage, we declare either $x <_t y$ or $y <_t x$. □

By Lemmas 3.1.2 - 3.1.5, the group structure of H is well-defined and H is a computable group. By Lemmas 3.1.6 - 3.1.7, the order $\leq_H = \bigcup_s \leq_s$ is computable. Since we have also satisfied the coding requirements, this concludes the proof of the theorem. □

3.2 Extending non-closure

In Chapter 1, we noted that if G is computable, abelian, and torsion-free with infinite rank then $\mathrm{Spec}(\mathbb{X}(G))$ has several properties not shared the degree spectra of arbitrary Π^0_1 classes. Among other things, it contains infinitely many low degrees, infinitely many hyperimmune-free degrees, and contains the cone above some c.e. degree. However, it cannot actually be a single cone of degrees (unless it contains every degree). In particular, it is not necessarily closed upwards in the Turing degrees:

Theorem 3.2.1 (Kach, Lange, & Solomon, 2013)**.** *There is a computable torsion-free abelian group G with infinite rank and a non-computable, computably enumerable set C such that G has exactly two computable orders, and every C-computable order on G is computable.*

In particular, the set of degrees of orders on this G is not closed upwards, because it is missing the entire lower cone below some incomplete c.e. set. A natural question is whether the set C can be made to have any other special properties. Certainly C cannot be $\mathbf{0}'$, since we know every such G has orders of degree $0'$. However, it is possible to ensure that C is a high set, i.e. $C \leq_T \mathbf{0}'$ but $C' \equiv_T \mathbf{0}''$.

Theorem 3.2.2. *There is a computable torsion-free abelian group G with infinite rank and a high c.e. set C such that G has exactly two computable orders, and every C-computable order on G is computable.*

Proof. This will be done using an infinite injury construction, organized via a tree of strategies. To construct the high set C, we use the standard method for doing so, namely using the limit lemma and the Π_2-complete index set Inf, attempting to build a Turing functional Γ to satisfy requirements

$$\mathcal{R}_e \ : \ \text{Inf}(e) = \lim_v \Gamma^C(e, v)$$

The rest of the construction will handle building the group G, defining the group operation $+_G$, defining a computable group order $\leq_G$, and attempting to satisfy requirements

$$\mathcal{S}_e \ : \ \text{if } \Phi_e^C \text{ is an order } \leq_e^C \text{ on } G\text{, then } \Phi_e^C \text{ is computable}$$

While the requirements above are very different than the coding requirements in the previous theorem, the construction of the group $(G, +_G)$ itself will be much the same as in the previous theorem. That is, we will construct G in stages with $G_0 \subseteq G_1 \subseteq G_2 \subseteq \ldots$ and $G = \bigcup_s G_s$. At stage s, we will have an approximate basis $\{b_0^s, b_1^s, \ldots, b_s^s\}$ with $b_0^s = 1$ and for each i, $b_i^t \neq b_i^{t+1}$ for at most one t. This will produce a basis for G consisting of elements $b_i = \lim_s b_i^s$ for each i. The elements of G_s will be assigned formal sums $g = q_0^s b_0^s + \ldots q_n^s b_n^s$ for some $n \leq s$ and some n-tuple of rational numbers $(q_0^s, q_1^s, \ldots, q_n^s)$. The formal sum assigned to $g \in G_s$ may change in order to meet the diagonalization requirements $\mathcal{S}_e$, but will reach a limit at some finite stage. The operation $+_G$ will be defined in stages as a partial binary relation on G_s such that $x +_s y = z$ for $x, y, z \in G_s$ implies $x +_t y = z$ in G_t for all $t \geq s$, and then defining $x +_G y = z$ in G if the limiting sums for x and y add to the limiting sum for z. As before, we will also use intervals $(a_e^s, \hat{a}_e^s)$ assigned to each b_e^s to define a computable order $\leq_G$ on G. To meet $\mathcal{S}_e$, we will show that if Φ_e^C is an order on G, then this order is actually equal to $\leq_G$ or $\leq_G^*$.

Construction: We begin as in the proof of the previous theorem. First, assign a priority order $\mathcal{R}_0 \prec \mathcal{S}_0 \prec \mathcal{R}_1 \prec \mathcal{S}_1 \prec \dots$, and fix a computable coding of $\mathbb{Q}^{<\omega}$. At stage 0, set $G_0 = \{0, 1\}$. We assign 0 the empty sum, so $0_G = 0$, and assign 1 the sum $1b_0^0$. Define $+_0$ to be the relation with $0_G +_0 0_G = 0_G$ and $0_G +_0 b_0^0 = b_0^0 +_0 0_G = b_0^0$. Define the approximate basis at stage 0 to be $\{b_0^0\}$.

More generally, at the end of stage s we will have a finite set G_s, an approximate basis $\{b_0^s, \dots, b_s^s\}$, and each $g \in G_s$ will be assigned a sum $g = q_0^s b_0^s + \dots q_n^s b_n^s$. As before, when convenient we can ensure $n = s$ by setting the terminal coefficients to zero. The partial function $+_s$ is defined on $(x, y) \in G_s^2$ if any only if there is a $z \in G_s$ such that the sums assigned to x and y add up componentwise to the sum assigned to z. For later stages $s + 1$, we will take one of two strategies, depending on whether we need to change an approximate basis element to diagonalize against Φ_e^C being an order on G which differs from $\leq_G$ and $\leq_G^*$. That is, we will do one of the following:

1. If the approximate basis is to be left unchanged, we set $b_i^{s+1} = b_i^s$ for all $i \leq s$. For each $g \in G_s$ (viewed as an element of G_{s+1}), we define $q_i^{s+1} = q_i^s$ and assign g the same sum with b_i^{s+1} and q_i^{s+1} replacing b_i^s and q_i^s. This ensures that if $x +_s y = z$ in G_s, then $x +_{s+1} y = z$ in G_{s+1}.

   We also add two new elements to G_{s+1}, one labeled b_{s+1}^{s+1} and assigned the sum $1b_{s+1}^{s+1}$, and the second given the sum $q_0^{s+1} b_0^{s+1} + \dots + q_n^{s+1} b_n^{s+1}$, where $\langle q_0^{s+1}, \dots, q_n^{s+1} \rangle$ is the least n-tuple of rationals with $n \leq s$ in our fixed enumeration that is not already assigned to an element of G_{s+1}.

2. If the approximate basis is to be altered in order to diagonalize, we will redefine an appropriately chosen b_j^s by adding a new dependency relation. For all $i \leq s$ with $i \neq j$, we set $b_i^{s+1} = b_i^s$. We will either set $b_j^s = q b_0^{s+1}$ for some rational q or

set $b_j^s = b_k^{s+1} + qb_0^{s+1}$ for some rational q and some index $k < j$. For each $g \in G_s$, we assign g the same sum, except we replace each b_i^s with b_i^{s+1} for $i \leq s$ and $i \neq j$, and replace b_j^s with one of the two expressions given above. For example, if we wish to declare $b_j^s = b_k^{s+1} + qb_0^{s+1}$, then an element of G_s assigned the sum $q_0^s b_0^s + \cdots + q_n^s b_n^s$ will be assigned (at stage $s+1$) the sum

$$\begin{aligned} & q_0^s b_0^{s+1} + \cdots + q_k^s b_k^{s+1} + \cdots + q_j^s (b_k^{s+1} + qb_0^{s+1}) + \cdots + q_s^s b_s^{s+1} \\ = {} & (q_o^s + q_j^s q) b_0^{s+1} + \cdots + (q_k^s + q_j^s) b_k^{s+1} + \cdots + 0 b_j^{s+1} + \cdots + q_s^s b_s^{s+1} \end{aligned}$$

Note that the approximate basis element b_j^{s+1} does not appear in the sum for any element of G_s viewed as an element of G_{s+1}, and the only approximate basis elements whose coefficients can change from zero to nonzero are b_0^s and b_k^s. Since $k < j$, this can only occur finitely often, and it follows that all of the approximate basis elements eventually reach a limit. This in turn implies that for any $g \in G_s$, there is a $t \geq s$ such that all of the coefficients q_i^t and b_i^t involved in the sum assigned to g in G_t will remain unchanged at all later stages $t' \geq t$. We also must ensure that the assignment of sums to elements of G remains injective when choosing the rational q. To do so, note that the diagonalization process to be specified later will place some restrictions on q, but we will see that there will always be infinitely many choices available. Additionally, if $x +_s y = z$ in G_s, by linearity we still have $x +_{s+1} y = z$ in G_{s+1}.

We also add three new elements to G_{s+1}, one labeled b_j^{s+1} and assigned the sum $1b_j^{s+1}$ (note that no other element of G_{s+1} has a nonzero b_j^{s+1} coefficient), one labeled b_{s+1}^{s+1} and assigned the sum $1b_{s+1}^{s+1}$, and a third assigned the sum $q_0^{s+1} b_0^{s+1} + \cdots + q_n^{s+1} b_n^{s+1}$, where $\langle q_0^{s+1}, \ldots, q_n^{s+1} \rangle$ is the least n-tuple of rationals with $n \leq s$ in our fixed enumeration that is not already assigned to an element of G_{s+1}, as before.

To define the computable order $\leq_G$, we use a similar technique to the previous theorem. The main change is that approximate basis elements may be mapped to rational multiples of $\sqrt{p_i}$ rather than just to $\sqrt{p_i}$. We define a Δ^0_2 map f from the set of approximate basis elements to the real numbers with each b^s_i assigned to a real $r^s_i = q\sqrt{p_i}$, a rational multiple of the i^{th} prime number. Precisely which rational multiple may change during the diagonalization process. To keep track of these, each b^s_i will also be assigned a rational interval $(a^s_i, \hat{a}^s_i)$ containing r^s_i with $\hat{a}^s_i - a^s_i < 2^{-s}$. Since the limit of each approximate basis element exists, the constraints ensure that the limit basis element b_i is mapped to a rational multiple of $\sqrt{p_i}$. As before, the interval constraints on each b^s_i impose an interval constraint on each $g \in G_s$ through the sum assigned to g. To define $\leq_s$ on G_s at stage s, we declare $x \leq_s x$ for each $x \in G_s$, and for each pair $x, y \in G_s$, we examine the interval constraints for each. If they are disjoint, we declare $x \leq_s y$ if the interval constraint for x lies entirely to the left of the interval constraint for y, we declare $y \leq_s x$ if the reverse is true, and otherwise we do not declare any ordering relation between x and y at stage s.

To maintain that $x \leq_s y$ implies $x \leq_{s+1} y$, it suffices to check that the interval constraint for x at stage $s+1$ is contained within the interval constraint for x at stage s. As before, at stage $s+1$, there are two possible cases.

1. Suppose we leave the approximate basis unchanged. Set $r^{s+1}_0 = 1$, and for each $i \leq s$, set $r^{s+1}_i = r^s_i$, and define $a^{s+1}_i, \hat{a}^{s+1}_i$ to be such that

$$r^{s+1}_i \in (a^{s+1}_i, \hat{a}^{s+1}_i)_{\mathbb{R}} \subseteq (a^s_i, \hat{a}^s_i)_{\mathbb{R}} \text{ and } \hat{a}^{s+1}_i - a^{s+1}_i < 2^{-(s+1)}$$

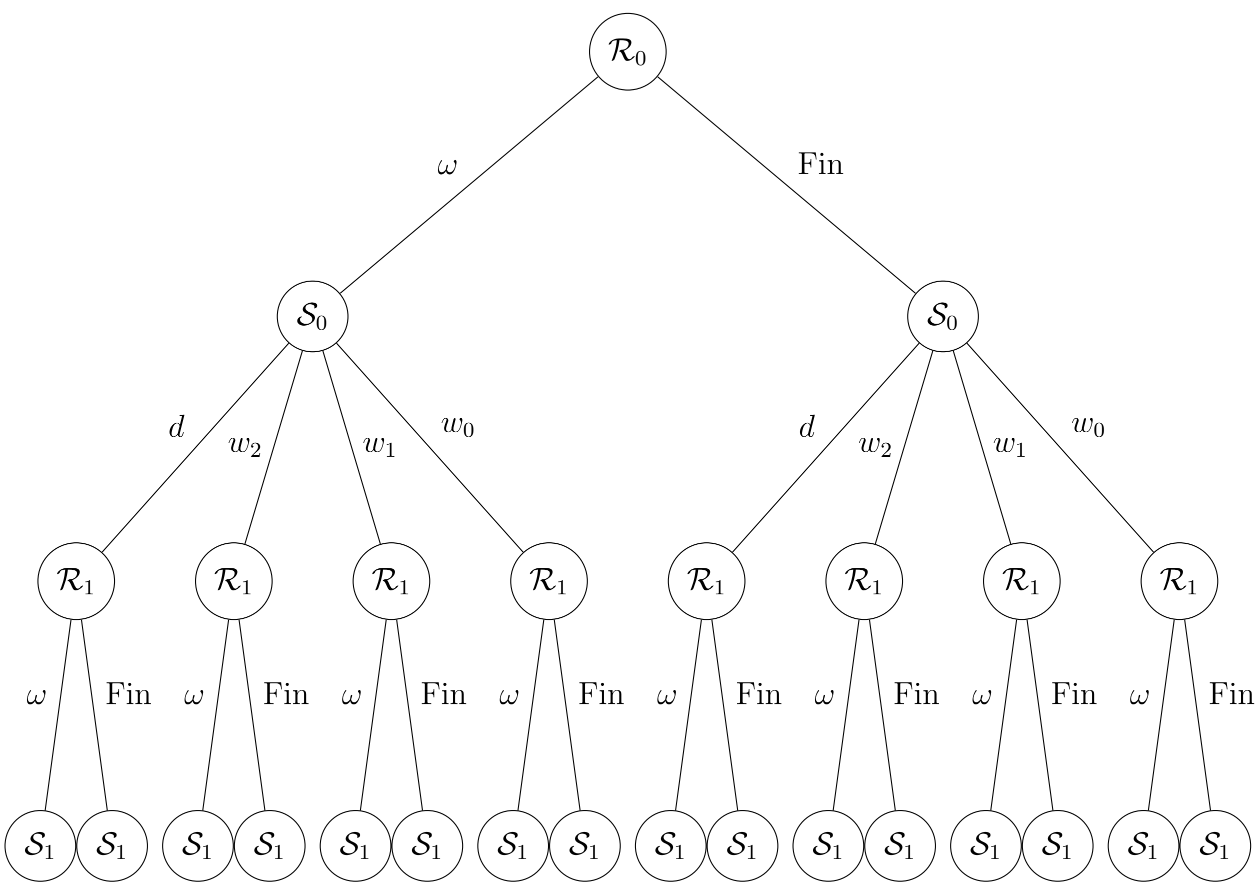
   Since the interval constraints for approximate basis elements are nested, the interval constrains for each $x \in G_s$ at stage $s+1$ are nested within the interval constraints at stage s. For the element b_{s+1}^{s+1} introduced at stage $s+1$, we define r_{s+1}^{s+1} to be a rational multiple of $\sqrt{p_{s+1}}$ and choose an interval $(a_{s+1}^{s+1}, \hat{a}_{s+1}^{s+1})$ containing r_{s+1}^{s+1} with length less than $2^{-(s+1)}$. The diagonalization process may place some constraints on which rational multiple of $\sqrt{p_{s+1}}$ is chosen. If no such restraints exist, we set $r_{s+1}^{s+1} = \sqrt{p_{s+1}}$.

2. If we add a dependency relation at stage $s+1$ and redefine b_j^s to be qb_0^{s+1} or $b_k^{s+1} + qb_0^{s+1}$, the ordering places some restrictions on the possible choices of which rational we use for q in the procedure above. There will be additional restrictions placed on q later for the sake of diagonalization, but all that is required for the general construction of the group is that there be infinitely many possible choices for q. For all the order relations declared by $\leq_s$ to hold at stage $s+1$, we need to ensure that the image of b_j^s at stage $s+1$ remains contained in the interval $(a_j^s, \hat{a}_j^s)_{\mathbb{R}}$. In the case when we set $b_j^{s+1} = qb_0^{s+1}$, we can choose any $q \in (a_j^s, \hat{a}_j^s)_{\mathbb{R}}$, since the image of b_0^{s+1} in $\mathbb{R}$ is 1, and there are clearly infinitely many such q.

   In the case when we set $b_j^{s+1} = b_k^{s+1} + qb_0^{s+1}$, we need to choose q and $(a_k^{s+1}, \hat{a}_k^{s+1})$ such that $(q + a_k^{s+1}, q + \hat{a}_k^{s+1})_{\mathbb{R}}$ is contained in $(a_j^s, \hat{a}_j^s)_{\mathbb{R}}$. Note that there are infinitely many choices of a_k^{s+1} and $\hat{a}_k^{s+1}$ that ensure that the interval between them has diameter less than $2^{-(s+1)}$. For the approximate basis elements b_i^{s+1} with $i \leq s$ and $i \neq j$, we proceed as in the case when no dependency relation is introduced, and for the new approximate basis elements b_j^{s+1} and b_{s+1}^{s+1}, we choose r_j^{s+1}, r_{s+1}^{s+1}, a_j^{s+1}, $\hat{a}_j^{s+1}$, a_{s+1}^{s+1}, and $\hat{a}_{s+1}^{s+1}$ accordingly.

We can now describe the tree of strategies for meeting the requirements $\mathcal{R}_0 \prec \mathcal{S}_0 \prec \mathcal{R}_1 \prec \mathcal{S}_1 \prec \ldots$. As noted at the beginning, $\mathcal{R}_e$ will be met by the standard high construction, and will have two outcomes, labeled ω and Fin, ordered as $\omega <_L$ Fin. The $\mathcal{S}_e$ requirements are finitary and will have four possible outcomes, labeled d, w_2, w_1, and w_0, ordered left to right as $d <_L \omega_2 <_L \omega_1 <_L \omega_0$. During the w_i outcomes, we wait for particular conditions to arise before proceeding to the next outcome on the left, preserving C up to some use whenever we do so. In the d outcome, we will diagonalize and satisfy $\mathcal{S}_e$ permanently. If α is a node on the tree at level $2e$, then α is an $\mathcal{R}_e$ strategy, and if α is at level $2e+1$ then it is an $\mathcal{S}_e$ strategy.

Overall structure of the tree of strategies:



Action for $\mathcal{R}_e$:

Let α be an $\mathcal{R}_e$ strategy which is eligible to act at stage s.

1. If s is the first stage at which α is eligible to act or the first stage at which it is eligible to act after being initialized, then set $s_\alpha = s$ and $n_\alpha = s$. Otherwise, leave these parameters with their previous values.

2. Check if there exists an $x \geq n_\alpha$ with $x \in W_{e,s}$.

   If NO, then set $\Gamma^{C_s}(e,s) = 0$ with large use $\gamma(e,s)$. Take outcome Fin.

   If YES, then for all v with $s_\alpha \leq v < s$, enumerate $\gamma(e,v)$ into C_s, and reset $\Gamma^{C_s}(e,v) = 1$ with use $\gamma(e,v)$. Set $\Gamma^{C_s}(e,s) = 1$ with large use $\gamma(e,s)$. Increment n_α and take outcome ω.

Note that if we have the YES answer in (2), then we may have already reset some computations to $\Gamma^{C_s}(e,v) = 1$. In that case, $\gamma(e,v)$ is already in C_{s-1}, so enumerating it into C again does nothing. That is, we only really change $\Gamma^{C_s}(e,v) = 1$ for v since the last time α reset computations.

Assuming there is a stage after which α is never initialized and is eligible to act infinitely often, α will either eventually always receive the NO answer in (2) if W_e is finite, in which case we will have $\lim_s \Gamma^C(e,s) = 0$, or it will receive the YES answer infinitely often, in which case it will reset all $\Gamma^C(e,s) = 1$ for all $s \geq s_\alpha$, in which case $\lim_s \Gamma^C(e,s) = 1$, and W_e will be infinite.

Action for $\mathcal{S}_e$:

Let α be an $\mathcal{S}_e$ strategy. We describe its action in isolation, but first we must discuss when α believes a computation of the form $\Phi_{e,s}^{C_s}(x)\downarrow$ in the context of a higher priority $\mathcal{R}_i$ requirement. To motivate why, suppose β is an $\mathcal{R}_i$ strategy with $\beta * \omega \subseteq \alpha$.

That is, β is a higher priority $\mathcal{R}_i$ strategy, and α believes β will enumerate infinitely many numbers into C.

Suppose α sees $\Phi_{e,s}^{C_s}(x)\downarrow$ with use u and wants to act based on this. The concern is that β will later enumerate an element below u into C, which destroys the computation α was relying on. To avoid this, α checks whether there is a number v such that $s_\beta \leq v \leq s$ with $\gamma(e,v) \leq u$ and $\gamma(e,v) \notin C$. If there is no such number, then β cannot interfere with the computation $\Phi_e^{C_s}(x)\downarrow$. If α finds such a $\gamma(e,v)$, then α knows that β will eventually put $\gamma(e,v)$ into C, destroying the computation, so α does not believe the computation.

From now on, whenever we say that α sees a computation $\Phi_e^{C_s}$ halt, we mean that this computation has a use such that for all $\mathcal{R}_i$ strategies β with $\beta * \omega \subseteq \alpha$, if $\gamma(e,v) \leq u$ and $s_\beta \leq v \leq s$, then $\gamma(e,v) \in C_s$. In practice, this ensures that higher priority $\mathcal{R}_i$ requirements cannot injure α.

We can now specify the action of an $\mathcal{S}_e$ strategy α which is eligible to act at stage s. In what follows, we will write $\leq_e^{C_s}$ in place of $\Phi_e^{C_s}$, with the understanding that $\Phi_e^{C_s}(x)\downarrow= 0$ means $x \leq_e^{C_s} 0_G$ and $\Phi_e^{C_s}(x)\downarrow= 1$ means $0_G \leq_e^{C_s} x$. To make the notation simpler, we also use $\leq_e^C$ for the order whose positive cone is given by Φ_e^C.

1. Wait for α to see a computation declaring $b_0 \leq_e^{C_s} 0_G$ or $0_G \leq_e^{C_s} b_0$.

   As long as this does not occur, α does not need to diagonalize by changing an approximate basis element, so wait and take outcome w_0. Note that if α takes outcome w_0 forever, then $\leq_e^C$ fails to be an order on G, and $\mathcal{S}_e$ is satisfied.

   If α sees a C-computation that eventually orders b_0 with respect to 0_G, take outcome w_1. The tree of strategies will ensure that C is preserved up to the use of this computation when $\mathcal{S}_e$ takes outcome w_1.

2. Suppose α has seen $0_G \leq_e^{C_s} b_0$. By restraining C, since $0_G \leq_G b_0$, we know that $\leq_e^C$ is not $\leq_G^*$, so it remains only to ensure that if $\leq_e^C$ is an order, then it is $\leq_e^G$. The case when α has seen a computation $b_0 \leq_e^{C_s} 0_G$ is exactly analogous, since we can work with $(\leq_e^C)^*$ instead, and show that if $(\leq_e^C)^*$ is an order, then it is equal to $\leq_G$.

   Wait until a stage s when we have approximate basis elements b_i^s and rationals $q_0 < q_1$ such that $q_0 b_0^s <_s b_j^s <_s q_1 b_0^s$ in G_s, but either $b_j^s <_e^{C_s} q_0 b_0^s$ or $q_1 b_0^s <_e^{C_s} b_j^s$. Until this occurs, wait and take outcome w_1. Note that if this never occurs and α takes outcome w_1 forever, then for every $i \geq 1$, $b_i = \lim_s b_i^s$ lies in the same rational cut relative to b_0 according to $\leq_e^C$ and $\leq_G$. But since the orderings $\leq_e^C$ and $\leq_G$ are generated by their respective relative orderings of basis elements, it follows that they must be the same order, and $\mathcal{S}_e$ will be satisfied with no further action required.

   When α sees such a stage, it still leaves the approximate basis unchanged, but ensures that b_{s+1}^{s+1} is contained in $(q_0 b_0^s, q_1 b_0^s)$ in stage $s+1$. To do so, it sets $r_{s+1}^{s+1} = q\sqrt{p_{s+1}}$ where $r_{s+1}^{s+1} \in (q_0, q_1)_{\mathbb{R}}$, and defines the sums assigned to elements of G_{s+1} as described above. It also chooses q and the interval $(a_{s+1}^{s+1}, \hat{a}_{s+1}^{s+1})$ to define the order on G_{s+1} so that

   (a) If $b_j^s \leq_e^C q_0 b_0^s$, then $b_{s+1}^{s+1} <_{s+1} b_j^{s+1}$

   (b) If $q_1 b_0^s \leq_e^C b_j^s$, then $b_j^{s+1} <_{s+1} b_{s+1}^{s+1}$

   This can be accomplished by choosing r_{s+1}^{s+1} to be a rational multiple of $\sqrt{p_{s+1}}$ for which there are appropriately close a_{s+1}^{s+1} and $\hat{a}_{s+1}^{s+1}$ satisfying either $\hat{a}_{s+1}^{s+1} < a_j^{a+1}$ in the first case, or $\hat{a}_j^{s+1} < a_{s+1}^{s+1}$ in the second.

After doing this, α ends the current stage of the construction. When α acts next, it will take outcome w_2. As above, this action causes C to be preserved up to the use of the computations α has seen thus far by the structure of the tree of strategies.

3. When α is next eligible to act, it waits for a stage t at which $\leq_e^C$ declares whether b_{s+1}^t is in $(q_0 b_0^t, q_1 b_1^t)$ or not. While waiting, continue to take outcome w_2. Again, note that if $\leq_e^C$ never specifies whether or not b_{s+1}^t is in this interval, it is not an order of G and $\mathcal{S}_e$ is satisfied. If α does eventually such a computation $\leq_e^{C_t}$, we act according to one of the following two cases.

   Suppose $\leq_e^{C_t}$ declares $b_{s+1}^t \notin (q_0 b_0^t, q_1 b_0^t)$. At stage $t+1$, we redefine b_{s+1}^t by adding the dependency relation $b_{s+1}^{t+1} = q b_0^{t+1}$ for a rational q. As described in the construction of the ordering, the rational q must be consistent with the current interval constraint for b_{s+1}^t, but since there are infinitely many such q (i.e. q in the interval $(a_{s+1}^t, \hat{a}_{s+1}^t)$), we may choose a q in this interval that keeps the assignment of sums to group elements injective. Furthermore, because the interval constraints are nested, $(a_{s+1}^t, \hat{a}_{s+1}^t) \subseteq (a_{s+1}^{s+1}, \hat{a}_{s+1}^{s+1}) \subseteq (q_0, q_1)$ in $\mathbb{R}$, which ensures that $q_0 < q < q_1$. Therefore, any ordering of G in which b_0 is positive must satisfy $q_0 b_0^t < q b_0^t < q_1 b_0^t$, and hence $q_0 b_0^t < b_{s+1}^t < q_1 b_0^t$. However, $\leq_e^C$ makes b_0 positive but does not satisfy this inequality. Therefore, by restraining C to preserve this computation, which we do by taking outcome d at all future stages where α is eligible to act, we have ensured that $\mathcal{S}_e$ is satisfied.

   On the other hand, suppose $\leq_e^{C_t}$ declares $b_{s+1}^t \in (q_0 b_0^t, q_1 b_0^t)$. We split into two subcases, depending on how C ordered b_j^s relative to $(q_0 b_0^s, q_1 b_0^s)$ in step 2 of the action of α.

(a) Suppose when we started this diagonalization process in step 2 at stage $s+1$, we saw $b_j^s \leq_e^C q_0 b_0^s$. In this case, we set $b_{s+1}^{s+1} <_{s+1} b_j^{s+1}$, so $b_{s+1}^t <_t b_j^t$. Since we restrained C at stage $s+1$ and $b_j^t = b_j^s$, we have $b_j^t \leq_e^C q_0 b_0^t$. At stage $t+1$, we restrain C one more time to preserve the fact that $b_{s+1}^t \in (q_0 b_0^t, q_1 b_0^t)_{\leq_e^C}$. It follows that if $\leq_e^C$ is an order, it must declare $b_j^t <_e^C b_{s+1}^t$.

We choose a positive rational q and add the dependency relation $b_{s+1}^t = b_j^{t+1} - q b_0^{t+1}$, leaving all other $b_i^{t+1} = b_i^t$. The rational q must be chosen so that our interval constraints for $\leq_{s+1}$ are nested inside the interval constraints for $\leq_s$. We can ensure this containment by choosing a_j^{t+1} and $\hat{a}_j^{t+1}$ so that $\hat{a}_j^{t+1} - a_j^{t+1} < \hat{a}_{s+1}^t - a_{s+1}^t$ as well as $\hat{a}_j^{t+1} - a_j^{t+1} < 2^{-(t+1)}$, and choose q such that $a_{s+1}^t < a_j^{t+1} - q < \hat{a}_j^{t+1} - q < \hat{a}_{s+1}^t$. Since $b_{s+1}^t <_t b_j^t$, and hence the interval $(a_{s+1}^t, \hat{a}_{s+1}^t)$ lies completely to the left of the interval $(a_j^t, \hat{a}_j^t)$, there are infinitely many positive q satisfying this condition.

To see that this action successfully diagonalizes against $\leq_e^C$, note that the relation $b_{s+1}^t = b_j^{t+1} - q b_0^{t+1}$ with q positive guarantees that any order $<$ on G which makes b_0^{t+1} positive must also make $b_{s+1}^t < b_j^{t+1}$. However, $\leq_e^C$ makes $b_0^{t+1} = b_0^t$ positive but is committed to making $b_j^{t+1} = b_j^t <_e^C b_{s+1}^t$. This is impossible, so $\leq_e^C$ is not an order of G and $\mathcal{S}_e$ is satisfied.

(b) On the other hand, suppose when we started the diagonalization at stage $s+1$, we saw $q_1 b_0^s \leq_e^C b_j^s$. In this case, we made $b_j^{s+1} <_{s+1} b_{s+1}^{s+1}$, so $b_j^t <_t b_{s+1}^t$. Restraining C to preserve the fact that $b_{s+1}^t \in (q_0 b_0^t, q_1 b_0^t)_{\leq_e^C}$ ensures that $b_{s+1}^t <_e^C b_j^t$. At stage $t+1$, we choose a positive rational q and add the dependency relation $b_{s+1}^t = b_j^{t+1} + q b_0^{t+1}$. As in the previous case, there are infinitely many such q we can use. This relation implies that any order which makes b_0 positive is committed to $b_j^{t+1} < b_{s+1}^t$. However, $\leq_e^C$ has set b_0 to be positive but has declared $b_{s+1}^t <_e^C b_j^t = b_j^{t+1}$, so again $\leq_e^C$ is not an order of G and $\mathcal{S}_e$ is satisfied.

In either of the above cases, when α acts at stage t, it defines the appropriate elements, order, and operation of G_t, and ends the stage. At all future stages, α takes outcome d. Note that this preserves the uses seen in 3.

Having specified the actions of both $\mathcal{R}_e$ and $\mathcal{S}_e$ strategies, we can now give the full construction. Stage 0 of the construction was already given. At stage $s+1$, we begin with the empty node λ, which is the only $\mathcal{R}_0$ node, and let strategies α on the tree act until either

1. an $\mathcal{S}_e$ strategy is reached which acts as in (2) or (3) above, in which case it defines G_{s+1} and ends the stage; or
2. a strategy α with $|\alpha| = s+1$ is reached, in which case we define G_{s+1} by leaving the approximate basis unchanged and setting $r_{s+1}^{s+1} = \sqrt{p_{s+1}}$, and we end the stage

Verification:

Let f_s denote the path in the tree of strategies taken at stage s of the construction. That is, $f_s = \alpha$, where α is the last node eligible to act at stage s. At the end of the stage, we initialize all strategies β such that β is a proper extension of α or α lies to the left of β on the tree.

Define the true path f to be the path through the tree of strategies such that $\alpha \in f$ if and only if there are infinitely many s such that $\alpha \subseteq f_s$ and there is some

t such that for all $s \geq t$ and all β with the same length as α lying to the left of α are not in f_s. To finish the proof of the theorem, we need to establish the following lemmas.

Lemma 3.2.3. *If α is on the true path, some successor of α is on the true path. That is, the true path f is infinite.*

Proof. Fix α on the true path and t such that no β with length $|\alpha|$ to the left of α is on f_s for any $s \geq t$. Without loss of generality, $t > |\alpha|$. If α is an $\mathcal{R}_e$ strategy and $\alpha \in f_s$ for $s \geq t$, then by construction either $\alpha * \omega \in f_s$ or $\alpha * \text{Fin} \in f_s$. If $\alpha * \omega \in f_s$ for infinitely many s, then $\alpha * \omega$ is on the true path. Otherwise, $\alpha * \text{Fin}$ is on the true path.

If α is an $\mathcal{S}_e$ strategy and $\alpha \in f_s$ for $s \geq t$, then α may act in such a way that ends the current stage. Note that it can do this at most once for each of (2) and (3) above, so we may assume that t is sufficiently large that if $\alpha \in f_s$ and $s \geq t$, then α does not end stage s. It follows that one of the finitely many outcomes of α is on f_s, and the leftmost outcome of α taken infinitely often is on the true path. □

Lemma 3.2.4. *Each approximate basis element b_i^s is redefined only finitely often, so the analogs of the lemmas presented in the previous theorem which ensure that $+_G$ and $\leq_G$ are well defined hold.*

Proof. The only approximate basis elements which are ever redefined during the construction are the elements b_{s+1}^{s+1} selected by an $\mathcal{S}_e$ strategy acting as in (2) above. Each such element can be chosen by at most one $\mathcal{S}_e$ strategy, and each $\mathcal{S}_e$ strategy acts to redefine this basis element at most once. □

Finally, we must verify that all the requirements are satisfied.

Lemma 3.2.5. *Each $\mathcal{R}_e$ requirement is met.*

Proof. Fix the $\mathcal{R}_e$ strategy α on the true path, and let s_0 be a stage such that α is never initialized after s_0. There are two possibilities.

1. If W_e is finite, then there is a stage $s_1 \geq s_0$ such that $W_e = W_{e,s_1}$. Then for all $s \geq s_1$, no new element enters W_e, so α never resets a computation after stage s_1. Furthermore, no other $\mathcal{R}_e$ strategy β resets a computation after s_1, so $\lim_s \Gamma^{C_s}(e, s) = 0 = \mathrm{Inf}(e)$, as required.

2. If W_e is infinite, there are infinitely many stages $s > s_0$ at which α is eligible to act and a number above n_α is in $W_{e,s}$. At each such stage, α resets $\Gamma^{C_s}(e, v) = 1$ for all $s_\alpha \leq v \leq s$. Therefore, for all $v \geq s_\alpha$, we have $\Gamma^{C_s}(e, v) = 1$, so $\lim_s \Gamma^{C_s}(e, s) = 1 = \mathrm{Inf}(e)$, as required.

□

Lemma 3.2.6. *Each $\mathcal{S}_e$ requirement is met.*

Proof. Fix the $\mathcal{S}_e$ strategy α on the true path, and let s be a stage such that α is never initialized after s. If $\alpha * w_0$ is on the true path, then $\leq_e^C$ does not order b_0 with respect to 0_G, hence is not an order of G and $\mathcal{S}_e$ is satisfied. If $\alpha * w_1$ is on the true path, then $\leq_e^C$ is exactly equal to $\leq_G$, and $\mathcal{S}_e$ is satisfied. If $\alpha * w_2$ is on the true path, then again $\leq_e^C$ fails to determine the relative order of some two group elements, hence is not an order and $\mathcal{S}_e$ is satisfied. Finally, if $\alpha * d$ is on the true path, then we have successfully diagonalized and forced $\leq_e^C$ to commit to contradictory ordering facts about b_j, so it is not an order and $\mathcal{S}_e$ is satisfied. □

This concludes the proof of the theorem. □

Interestingly, note that in the above proof we do not have to ensure that C is not complete, as in the standard construction of an incomplete high set. This set C cannot have degree $\mathbf{0}'$, or $\deg(\mathbb{X}(G))$ would not contain infinitely many low degrees.

Chapter 4

Further Results

4.1 Randomness

Since defining an order on a group is equivalent to making a countable sequence of yes/no decisions, a natural question to ask is what happens if all of those decisions were made at random. Recall that an infinite binary sequence $S \in 2^\omega$ is Martin-Löf random (or 1-random) if it is not contained in the intersection of any uniform sequence $\{U_n\}_{n\in\omega}$ of Σ^0_1 classes with $\mu(U_n) \leq 2^{-n}$ for each n.

Equivalently, S is 1-random if for all n, the prefix-free Kolmogorov complexity of $S \upharpoonright n$ is within a constant of n. However, the algebraic structure of groups is such that the resulting order relation is always non-random, even if the order was defined by some random process.

Theorem 4.1.1. *Let G be a computable torsion-free abelian group. If P is the positive cone of an order on G, then P is not Martin-Löf random.*

Proof. Let P be the positive cone of $\leq$. We construct a Martin-Löf test $\{U_n\}$ with $P \in \bigcap_{n\in\omega} U_n$ as follows. Fix a 1-1 enumeration $G = \{0_G, g_1, g_2, \ldots\}$ of G, and let $U_0 = 2^\omega$. Suppose $g_1 > 0$. As elements g_i are enumerated into G, wait until the element $2g_1$ appears, say $2g_1 = g_s$. Let $U_1 = \{\tau \in 2^\omega : \tau \supseteq \sigma 1, |\sigma| = s - 1\}$. Note that $\mu(U_1) \leq \frac{1}{2}$. Now choose n to be the least positive integer such that $g_i \neq ng_1$ for all $i \leq s$, wait until an element $g_t = ng_1$ is enumerated into G, let $U_2 = \{\tau \in U_1 : \tau \supseteq \sigma 1, |\sigma| = t - 1\}$, and note that since $t > s$, $\mu(U_2) \leq \frac{1}{4}$.

Continuing in this way produces the desired sets U_n, each with measure at most 2^{-n}. $\{U_n\}$ is uniformly Σ^0_1 because each U_i is actually computable. The elements of $\bigcap_{n\in\omega} U_n$ are all $S \in 2^\omega$ such that $S(k_i) = 1$ for some fixed infinite sequence (k_i) satisfying $g_{k_i} = n_i g_1$ for our chosen sequence of multiples n_i. If $g_1 > 0$, then P is such an S, since all such $g_{k_i} > 0$. If $g_1 < 0$, replace the condition in the definition of the U_n with extending nodes of the form $\sigma 0$ instead of nodes of the form $\sigma 1$. $\square$

Martin-Löf randomness can also be categorized in terms of martingales. A martingale is a function $d : 2^{<\omega} \to \mathbb{R}^{\geq 0}$ satisfying $d(\sigma) = \frac{1}{2}(d(\sigma 0) + d(\sigma 1))$ for all σ, and one can show that S is Martin-Löf random if and only if no c.e. martingale d has $\limsup_n d(S \upharpoonright n) = \infty$. Intuitively, this is meant to capture the idea it is impossible to acquire unlimited capital by betting on whether $n \in S$ while following an effective strategy. In the case of group orders, suppose we wanted to place bets on whether group elements would be positive, watching them be enumerated into G one at a time. A simple (computable) strategy to do so mirrors the structure of the proof above. Place an arbitrary bet on g_1, and then simply wait until an element of the form $2g_1$ is enumerated into G, wagering nothing on all of the intermediary elements, and then wagering all capital on the certain bet that $2g_1$ has the same sign as g_1.

From there, let $n \in \mathbb{N}$ be the least such that ng_1 is not yet enumerated into G, abstain from betting until ng_1 is enumerated into G, and repeat this process indefinitely.

This can be summarized simply by observing that even though in an computable torsion-free abelian group of infinite rank it is possible to define an ordering by choosing a path through 2^ω uniformly at random (corresponding to the sign of each basis element), it is not possible for the entire ordering to be effectively random, because each fact about the ordering determines infinitely many other facts through the algebraic structure of the group. However, while $\mathbb{X}(G)$ contains no elements which are themselves random, it does contain elements with random degree, since every degree above $\mathbf{0}'$ is the degree of a 1-random set.

4.2 Thin Classes

The idea behind the above proof can also be used to show the following:

Theorem 4.2.1. *Let $\mathbb{X}(G)$ be the space of orders on a computable torsion-free group with infinite rank. There is a Π^0_1 subclass $\mathcal{C}$ of $\mathbb{X}(G)$ with no computable elements.*

Proof. To set up the diagonalization strategy for this proof, fix two independent elements a, b of G and let $\{a, b, c_0, c_1, c_2, \dots\}$ be a basis for G. It will be convenient below to know that orders of G with a particular form exist, and we need only have these orders classically. That is, we temporarily ignore all computability considerations.

The orders we will be concerned with are defined as follows. Let $\{p_i\}_{i\geq 0}$ enumerate the prime numbers, and let $\leq_{\mathbb{R}^2}$ be the order defined on $\mathbb{R}^2$ by

$$(u, v) \leq_{\mathbb{R}^2} (x, y) \Longleftrightarrow (v <_{\mathbb{R}} y) \vee (v = y \wedge u \leq_{\mathbb{R}} x)$$

That is, $\leq_{\mathbb{R}^2}$ is the reverse lexicographic order on $\mathbb{R}^2$. Under this order, $\mathbb{R}^2$ has two nontrivial archimedean classes. Namely, the elements of the form $(u, 0)$ with $u \neq 0$ and the elements of the form (x, y) with $y \neq 0$, and $(u, 0)$ is archimedean less than (x, y) in this case.

Fix some irrational real number r with $\frac{1}{2} < r < 1$. We will use r and $\{p_i\}$ to define an order $\leq_r$ on G. In particular, define a map $f : \{a, b, c_0, c_1, \dots\} \to \mathbb{R}^2$ by $a \mapsto (1, 0)$, $b \mapsto (r, 0)$, and $c_i \mapsto (0, \sqrt{p_i})$. This function extends linearly to an injective homomorphism $f : G \to \mathbb{R}^2$. Let $\leq_r$ be the order defined by $g \leq_r h$ if and only if $f(g) \leq_{\mathbb{R}^2} f(h)$. Intuitively, $\leq_r$ orders a and b by sending a to 1 and b to r in the smaller archimedean class of $(\mathbb{R}^2, \leq_{\mathbb{R}^2})$, and each element generated by $\{c_0, c_1, \dots\}$ is sent to the larger archimedean class.

Construction: We can now specify the construction of our Π^0_1 subclass $\mathcal{C}$ of $\mathbb{X}(G)$ with no computable elements. The orders $\leq_r$ described above will be used to ensure that $\mathcal{C}$ is nonempty. We build $\mathcal{C}$ to meet requirements

$$\mathcal{R}_e : \Phi_e \text{ is not an order on } G \text{ in } \mathcal{C}$$

The overall strategy will be to define an effective sequence of Π^0_1 classes $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \dots$ contained in $\mathbb{X}(G)$ such that $\mathcal{C} = \bigcap_s \mathcal{C}_s$ is nonempty and meets each requirement $\mathcal{R}_e$. Since an effective intersection of Π^0_1 classes is a Π^0_1 class, this will be the desired Π^0_1 subclass $\mathcal{C}$.

At stage 0, set $\mathcal{C}_0 \subseteq \mathbb{X}(G)$ to be the set of all $P \in \mathbb{X}(G)$ such that P declares $0 < b < a < 2b$. That is, $b \in P$, $a - b \in P$, and $2b - a \in P$. Note that $\mathcal{C}_0$ contains all orders of the form $\leq_r$ as described above. The intuition for meeting $\mathcal{R}_e$ is as follows. If Φ_e defines an order $\leq_e$ in $\mathcal{C}_0$, it has to eventually narrow down the relative positions of a and b so b lies within some rational interval $q_0 a <_e b <_e q_1 a$, where $0 < q_1, q_2 < 1$ are rational. In particular, eventually we can find such an interval with $q_1 - q_0$ arbitrarily small, and once we find a suitable interval, we will remove all orders that satisfy this pair of inequalities from $\mathcal{C}_0$. To be more precise, notice that the rational multiple $q_0 a$ need not be a member of G. If $q_0 = \frac{m_0}{n_0}$, then by $q_0 a < b$ we literally mean $n_0 b - m_0 a$ is positive, and that involves only elements of G. However, for clarity we continue to speak of rational multiple inequalities in what follows, with the understanding that they are to be interpreted as the equivalent integer multiple inequalities.

Strategy for $\mathcal{R}_e$: Partition $(\frac{1}{2}, 1)$ into equal subintervals of length $2^{-(e+3)}$, and wait until Φ_e halts on a computation that places b in aI for one such subinterval I. To make this precise, note that to say Φ_e declares $b \in (\frac{m_0}{n_0}a, \frac{m_1}{n_1}a)$ really means that Φ_e is the positive cone of an order with $n_0 b - m_0 a > 0$ and $m_1 a - n_1 b > 0$, so $\Phi_e(n_0 b - m_0 a) \downarrow = \Phi_e(m_1 a - n_1 b) \downarrow = 1$. If this occurs at stage s, we can force b to not lie in this interval by deleting all extensions of nodes σ satisfying both $\sigma(i_0) = \sigma(i_1) = 1$, where $n_0 b - m_0 a = g_{i_0}$ and $m_1 a - n_1 b = g_{i_1}$. In other words, we set $\mathcal{C}_{s+1} = \{P \in \mathcal{C}_s : P \text{ satisfies } b \notin (\frac{m_0}{n_0}a, \frac{m_1}{n_1}a)\}$. If this never occurs, then Φ_e cannot be an order on G.

To see that the resulting $\mathcal{C}$ is nonempty, consider the possible position of b relative to a in the intersection. For each requirement $\mathcal{R}_e$, we remove at most one interval of length $2^{-(e+3)}$ from $(\frac{1}{2}, 1)$ in which b can lie relative to a. Therefore, we are left with a subset of $(\frac{1}{2}, 1)$ of measure at least $\frac{1}{4}$ which is not removed at any stage. Let r be any irrational real with $\frac{1}{2} < r < 1$ such that r is not in any interval removed during the construction. Then the order $\leq_r$ as described above lies in $\mathcal{C}$, and it follows that $\mathcal{C}$ is a nonempty subclass of $\mathbb{X}(G)$ with no computable member.

□

Several results follow immediately from this theorem. In particular, there are many results known about Π^0_1 classes with no computable elements, which are often called special Π^0_1 classes. For example,

Corollary 4.2.2. *Let $\mathbb{X}(G)$ be the space of orders on a computable torsion-free group with infinite rank. Then*

1. *$\mathbb{X}(G)$ contains elements with infinitely many different low degrees*
2. *$\mathbb{X}(G)$ contains elements with infinitely many different hyperimmune-free degrees*
3. *For any countable set of degrees $\{\mathbf{a}_i : i \in \omega\}$, $\mathbb{X}(G)$ contains an element which is incomparable with all $\mathbf{a}_i$*
4. *There exist elements $P, Q \in \mathbb{X}(G)$ such that $\deg(P) \wedge \deg(Q) = \mathbf{0}$*

Proof. All of the above are known to be true of every special Π^0_1 class, and applying this to the special Π^0_1 subclass $\mathcal{C}$ of $\mathbb{X}(G)$ gives the corollary. □

In Chapter 1, we mentioned Corollary 4.2.2 (1) as a result of Kach, Lang, Solomon, and Turetsky. Their proof was different, dealing with the structure of $\mathbb{X}(G)$ directly, and Theorem 4.2.1 gives a more complete picture of why this result holds.

An argument similar to that used in the previous theorem can be used to show that no Π^0_1 class of the form $\mathbb{X}(G)$ can be thin. Recall that a Π^0_1 class $\mathcal{P}$ is thin if every Π^0_1 subclass $\mathcal{C} \subseteq \mathcal{P}$ is of the form $\mathcal{C} = \mathcal{P} \cap S$ for some clopen set S, which is a union of finitely many cones in $\mathcal{P}$. In both this and the previous result, we make essential use of the fact that no finite collection of ordering facts is can fully specify the relative ordering of two independent elements of G.

Proposition 4.2.3. *Let $\mathbb{X}(G)$ be the space of orders on a computable torsion-free group with infinite rank. Then $\mathbb{X}(G)$ is not a thin Π^0_1 class.*

Proof. We need to construct a nonempty Π^0_1 subclass $\mathcal{C}$ of $\mathbb{X}(G) = [T]$ such that $\mathcal{C} \neq X(G) \cap \mathcal{D}$ for any clopen set $\mathcal{D} \subseteq 2^{<\omega}$. Note that each such $\mathcal{D}$ has the form $\bigcup_{i \leq n}[\sigma_i]$ for some finite set of nodes σ_i in $\mathbb{X}(G)$, and distinct σ_i may be assumed to be incomparable. Fix an enumeration $\sigma_0, \sigma_1, \ldots$ of $2^{<\omega}$, and define requirements $\mathcal{R}_i$ for each $i \geq 0$ as

$$\mathcal{R}_i : \text{If } \mathbb{X}(G) \cap [\sigma_i] \neq \emptyset, \text{ then } \mathcal{C} \cap [\sigma_i] \neq \mathbb{X}(G) \cap [\sigma_i]$$

To see why this suffices, note that $\mathbb{X}(G) \cap \left(\bigcup_{i \leq n}[\sigma_{j_i}]\right) = \bigcup_{i \leq n} (\mathbb{X}(G) \cap [\sigma_{j_i}])$. If $\mathcal{R}_e$ is met, since $\mathcal{C} \subseteq \mathbb{X}(G)$ it must be the case that $\mathcal{C}$ does not contain some extension of σ_e in $\mathbb{X}(G)$, if any such extensions exist. In particular, if each $\mathcal{R}_e$ is met, then any nonempty finite union of cones in $\mathbb{X}(G)$ must a strict superset of $\mathcal{C}$.

Construction:

Fix an enumeration $G = \{g_0, g_1, \ldots\}$. We build $\mathcal{C} = \bigcap_{s \in \omega} \mathcal{C}_s$, and will verify later that the resulting $\mathcal{C}$ is nonempty. Fix two independent elements $a, b \in G$, and at stage 0, let $\mathcal{C}_0 \subseteq \mathbb{X}(G)$ be the set of orders which declare $0 < b < a < 2b$. Since a, b are independent, there are infinitely many such orders which place b in the interval $(\frac{a}{2}, a)$. At each later stage $s + 1 \geq 1$, we work to satisfy $\mathcal{R}_s$ as by performing the following steps to define $\mathcal{C}_{s+1}$:

1. Choose a large n_s such that $n_s > s$ and if $x < |\sigma_s|$ and $g_x = kb$, then $k < n_s$.

2. Divide the interval $(\frac{a}{2}, a)$ into subintervals of length $2^{-(n_s+3)}$. Compute levels of T until a $T_{\hat{s}}$ is found such that for each $\tau \in T_{\hat{s}}$ with $|\tau| = \hat{s}$ and $\tau \supseteq \sigma_s$, τ has

assigned b to lie in one of the length $2^{-(n_s+3)}$ subintervals of $(\frac{a}{2}, a)$. Note that since every $\tau \supseteq \sigma_s$ in T which extends to an order on G must eventually either assign b to one such subinterval, by compactness we will either eventually find such a $T_{\hat{s}}$, or $[T] \cap [\sigma_s] = \emptyset$. In the latter case, we declare $\mathcal{R}_s$ won and proceed to (4) below. Otherwise, proceed to (3).

3. Suppose we have found a level $T_{\hat{s}}$ as previously described, and let $\{\tau_0^s, \tau_1^s, \ldots, \tau_{k_s}^s\}$ denote the corresponding length $\hat{s}$ extensions of σ in $T_{\hat{s}}$. To attempt to ensure that $C \cap [\sigma_s] \neq [T] \cap [\sigma_s]$, we remove $[\tau_0^s]$ from $\mathcal{C}$, set $r_s = 0$ (a parameter which will record the index of the τ_i^s whose extensions were last removed from $\mathcal{C}$), and proceed to (4). Note that if τ_0^s actually extends to a path in $[T]$, then $\mathcal{C}$ will fail to contain this path, and $\mathcal{R}_s$ will be met. Otherwise, we will eventually see a level $s^* > \hat{s}$ where τ_0^s has no extension in T_{s^*}. This is problematic, since it would mean that removing $[\tau_0^s]$ from $\mathcal{C}$ did not actually remove any elements of $[T] = \mathbb{X}(G)$ and $\mathcal{R}_s$ may no longer be satisfied. The purpose of (4) below is to deal with this possibility for $\mathcal{R}_t$ with $t < s$.

4. For each $t < s$, if $\mathcal{R}_t$ has not yet been won, compute whether $[\tau_{r_t}^t] \cap T_s = \emptyset$.

   If NO, then no further action is required on the part of $\mathcal{R}_t$ at this stage.

   If YES, then we must act to ensure that $\mathcal{R}_t$ remains satisfied, since our earlier action of removing $[\tau_{r_t}^t]$ from $\mathcal{C}$ was not sufficient. There are two cases to consider. If $r_t = k_t$, then all extensions of σ_t have terminated at level s, so $[T] \cap [\sigma_t] = \emptyset$ and we declare $[T] \cap [\sigma_t]$ won and do not alter $\mathcal{C}_s$ on behalf of $\mathcal{R}_t$. Otherwise, $r_t < k_t$, in which case we remove $[\tau_{r_t+1}^t]$ from $\mathcal{C}_s$ and set r_t to be $r_t + 1$. To end stage $s + 1$, set $\mathcal{C}_{s+1}$ to be equal to $\mathcal{C}_s$ with the indicated cones removed.

Verification:

Define $\mathcal{C}$ to be $\bigcap_s \mathcal{C}_s$. The above construction shows that $\mathcal{C}$ is a Π^0_1 subclass of $\mathbb{X}(G)$, so it remains to show that $\mathcal{C}$ is nonempty and each $\mathcal{R}_s$ is satisfied.

Lemma 4.2.4. *Each $\mathcal{R}_s$ requirement is met.*

Proof. Suppose $\mathbb{X}(G) \cap [\sigma_s] \neq \emptyset$, and let $\{\tau^s_0, \tau^s_1, \ldots, \tau^s_{k_s}\}$ be the extensions of σ_s denoted in (3) above. By assumption, σ_s extends to a path in $[T]$. Let P be the leftmost such path, and let τ^s_j be the node in $T_{\hat{s}}$ which lies on P. Since none of the τ^s_i with $i < j$ extend to paths in $[T]$, the cones above those nodes were removed from $\mathcal{C}$ in (2) or (4) above, and the last time we received a YES answer in (4), $[\tau^s_j]$ was removed from $\mathcal{C}$. In particular, $P \in \mathbb{X}(G) \cap [\sigma_s]$ but $P \notin C \cap [\sigma_s]$, so $\mathcal{R}_s$ is met. □

Lemma 4.2.5. *The Π^0_1 class $\mathcal{C}$ is nonempty.*

Proof. As noted previously, $\mathcal{C}_0$ is nonempty. If $\mathcal{R}_i$ was ever declared won in (2) or (4) above, then all of the cones we removed from $\mathcal{C}$ on account of $\mathcal{R}_i$ were actually empty in $\mathbb{X}(G)$, so the only concern is the $\mathcal{R}_s$ which were not explicitly won. For each such s, let j_s be such that $[\tau^s_{r_{j_s}}]$ was the last cone removed from $\mathcal{C}$ by $\mathcal{R}_s$, and let I_s be the corresponding interval of length $2^{-(n_s+3)}$. That is, σ_s extends to an order which places b in I_s. Furthermore, $\mathcal{C} = \mathbb{X}(G) \setminus \bigcup_s [\tau^s_{r_{j_s}}]$ with $\mathcal{R}_s$ not explicitly won and j_s defined as above.

Define a closed, but not necessarily effectively closed set $\hat{\mathcal{C}}$ by

$$\hat{\mathcal{C}} = \mathbb{X}(G) \setminus \{P : P \text{ places } b \text{ into } I_s \text{ for some } s \text{ with } \mathcal{R}_s \text{ not explicitly won}\}.$$

Note that $\hat{\mathcal{C}} \subseteq \mathcal{C}$, so it suffices to show that $\hat{\mathcal{C}}$ is nonempty. The collection of intervals I_t removed in this way from $(\frac{a}{2}, a)$ have total length at most $\sum_{t \geq 0} 2^{-(t+3)} = \frac{1}{4}$, so we can select an irrational $r \in (\frac{1}{2}, 1)_{\mathbb{R}}$ which is not contained in any of these I_t. Let $\{a, b, c_0, c_1, c_2, \dots\}$ be a basis for G. Defining a map $f : G \to R$ with $f(a) = 1$, $f(b) = r$, and $f(c_i) = \sqrt{p_i}$ for $i \geq 0$, extending f via linearity to the rest of G. As in previous results, such a map induces an order $\leq_r$ on G by declaring $g_1 \leq_r g_2$ in G if and only if $f(g_1) \leq f(g_2)$ in the standard ordering of $\mathbb{R}$. This order is not computable in general, but by construction, $\leq_r$ is an element of $\mathbb{X}(G)$ which is contained in $\hat{\mathcal{C}}$, hence is contained in $\mathcal{C}$. □

Since the Π^0_1 class $\mathcal{C}$ cannot have the form $\mathbb{X}(G) \cap \mathcal{D}$ for any clopen set $\mathcal{D}$, $\mathbb{X}(G)$ cannot be a thin Π^0_1 class. □

The fact that $\mathbb{X}(G)$ cannot be thin also follows from a result of Cenzer, Downey, Jockusch, and Soare. In particular, the authors show that if P is a member of a thin Π^0_1 class, then $P' \leq_T P + 0''$. On the other hand, we know that $\mathbb{X}(G)$ always contains all degrees above $\mathbf{0}'$. Taking $P \in \mathbb{X}(G)$ to be an order on G with degree $\mathbf{0}''$, we have $P' \equiv_T (0'')' \equiv_T 0'''$, but $P + 0'' \equiv_T 0'' + 0'' \equiv_T 0''$. However, it is interesting that this can be shown directly using the structure of G.

Bibliography

[1] Cenzer, D., Downey, R., Jockusch, C. G., and R. I. Shore, (1993). "Countable thin Pi^0_1 classes," *Ann. Pure Appl. Logic*, 59: pp. 79–139

[2] Cenzer, D. and J. Remmel, (1999). "Π^0_1 classes in mathematics," in Y. Ershov, S. Goncharov, A. Nerode and J. Remmel, eds., *Handbook of recursive mathematics, Volume 2*, pp. 623–821, North Holland, Amsterdam.

[3] Dzhafarov, D. and C. G. Jockusch, Jr., (2009). "Ramsey's theorem and cone avoidance," *J. Symbolic Logic*, 74 (2): pp. 557–578.

[4] Dobritsa, V., (1983). "Some constructivizations of abelian groups," *Siberian Math. J.*, 24 (2): pp. 167-173.

[5] Dabkowska, M. A., Dabkowski, M. K., Harizanov, V. S., and A. A. Togha (2010). "Spaces of orders and their Turing degree spectra", *Ann. Pure Appl. Logic*, 161, pp. 1134–1143.

[6] Downey, R. and D. Hirschfeldt, (2007). *Algorithmic Randomness and Complexity*, Springer–Verlag, New York.

[7] Downey, R. and S. A. Kurtz, (1986). "Recursion theory and ordered groups," *Ann. Pure Appl. Logic*, 32: pp. 137-151.

[8] Fuchs, L. (1963). *Partially ordered algebraic systems*, Pergamon Press, Oxford.

[9] Jockusch Jr., C. G. and R. I. Soare, (1972). "Π^0_1 classes and degrees of theories," *Trans. Amer. Math. Soc*, 173: pp. 33-56.

[10] Kach, A., Lange, K., and R. Solomon, (2013). "Degrees of orders on torsion-free abelian groups", *Ann. Pure Appl. Logic*, 164 (7-8): pp. 822–836.

[11] Kokorin, A. and V. Kopytov, (1974). *Fully ordered groups*, Halsted Press, New York.

[12] Levi, F. W., (1942). "Ordered groups," *P. Indian Acad. Sci. A*, 822 (16), pp. 256–263.

[13] Metakides, G. and A. Nerode (1979). "Effective content of field theory," *Ann. Math. Logic*, 17: pp. 289-320.

[14] Rabin, M. (1960). "Computable algebra, general theory and theory of computable fields," *T. Am. Math Soc.*, 95: pp. 341–360.

[15] Soare, R. (1989). *Recursively enumerable sets and degrees*, Springer-Verlag, Berlin.

[16] Solomon, R. (1998). "Reverse mathematics and fully ordered groups," *Notre Dame J. Formal Logic*, 39 (2): pp. 157-189.

[17] Solomon, R. (2002). "Π^0_1 classes and orderable groups," *Ann. Pure Appl. Logic*, 115: pp. 279–302.

[18] Teh, H. (1960). "Construction of orders in abelian groups," *Proc. Cambridge Philo. Soc.*, 57: pp. 476-482.