

LIMITWISE MONOTONIC FUNCTIONS AND THEIR APPLICATIONS

RODNEY G. DOWNEY, ASHER M. KACH, AND DANIEL TURETSKY

ABSTRACT. We survey what is known about limitwise monotonic functions and sets and discuss their applications in effective algebra and computable model theory. Additionally, we characterize the computably enumerable degrees that are *totally limitwise monotonic*, show the support strictly increasing $\mathbf{0}'$ -limitwise monotonic sets on \mathbb{Q} do not capture the sets with computable strong η -representations, and study the limitwise monotonic spectra of a set.

1. INTRODUCTION

Early applications of computability theory for demonstrating that various processes in mathematics were algorithmically unsolvable tended to be rather crude codings of the halting problem into the relevant mathematical structure. A classical example is the Novikov-Boone proof of the undecidability of the word problem in finitely presented groups (see [3], [4], [5], [6], and [7] and [25]). In that proof, a finitely presented group is constructed around a given description of the quadruples of a Turing machine, as in the proof of the undecidability of the word problem for finitely presented semigroups (see [27]), and then algebra is used to make the machine's action faithfully represented in the group. As observed by Post (see [26]), in many contexts it is enough to simply have the halting problem as a set and then code, such as the proof of Gödel's incompleteness theorem.

Computable model theory, and later reverse mathematics, pointed at encodings which were more complex. In computable model theory, traditionally we assume that we are given some structure whose elements are coded by the integers and whose open diagram is computable. In particular, a computable linear ordering would simply be $(A : \leq)$, where the universe A is computable and the ordering \leq is a computable relation on $A \times A$. Orderings are a natural arena to find codings other than the halting problem, since it is quite hard to code sets into them at low levels since little is definable with one quantifier. One of the first applications of more complex codings was due to Feiner (see [12]) who demonstrated how to code a Σ_3^0 set into a computable linear ordering via sizes of *finite maximal blocks*, i.e., a finite collection of points, all adjacent, such that the left and right endpoints are limit points. One of the classical applications of Feiner's Theorem is to construct a $\mathbf{0}'$ -computable linear ordering not classically isomorphic to a computable one by applying this result in relativized form and noting that there are sets S which are $\Sigma_3^{\mathbf{0}'}$ -computable but not Σ_3^0 -computable.

Key words and phrases. limitwise monotonic function.

The first and second author's research was supported by The Marsden Fund of New Zealand, the latter via a Post-Doctoral Fellowship. The third author's research was supported by a VIGRE Fellowship and a Research Assistantship from the University of Wisconsin Graduate School.

Later, Lerman studied such codings where the finite maximal blocks were separated by the order type ζ .

Definition 1.1. The *strong ζ -representation* of a set $S = \{n_0 < n_1 < n_2 < \dots\}$ is the linear order

$$\zeta + n_0 + \zeta + n_1 + \zeta + n_2 + \dots$$

A *weak ζ -representation* of a set $S = \{n_0 < n_1 < n_2 < \dots\}$ is a linear order

$$\zeta + n_{f(0)} + \zeta + n_{f(1)} + \zeta + n_{f(2)} + \dots$$

for some (total) surjective function f .

Theorem 1.2 (Lerman [24]). *A set S has a computable strong ζ -representation if and only if S is Δ_3^0 . A set S has a computable weak ζ -representation if and only if S is Σ_3^0 .*

There are many many applications of the technique of coding higher level sets into algebraic invariants of some algebraic object. They include applications in (abelian) group theory, ring theory, logic, lattice theory, etc. Here we refer the reader to [2] for such examples.

Beginning with the work of Khisamiev (see [20]), more subtle considerations came into such codings when it was realized that *not only the arithmetical complexity of the set is important, but also the manner of the formation of the set*. Khisamiev's intuition was that computability is concerned with dynamic enumerations, and this fact has ramifications for objects in computable structures. Khisamiev's example was in abelian p -groups, and we will look at his example later, but the nature of our concern is best illustrated by asking which equivalence structures are computable. Note that if we have a computable equivalence relation \equiv , then $[x]_{\equiv}$, the equivalence class of x , has the following important property: *it only gets bigger!* That is, at the stage that x enters the universe of the relation, we might discover that $[x]$ has n many elements. At later stages, the class can only gain elements. This phenomenon is captured by limitwise monotonicity.

Definition 1.3. A function F is *limitwise monotonic* if there is a computable approximation function $f(\cdot, \cdot)$ such that, for all x ,

- (i) $F(x) = \lim_s f(x, s)$.
- (ii) For all s , $f(x, s) \leq f(x, s + 1)$.

A set S is *limitwise monotonic* if it is the range of a limitwise monotonic function.

It is a very easy argument we leave to the reader to prove the following:

Theorem 1.4 (Calvert, Cenzer, Harizanov, and Morozov [8]). *An equivalence structure \mathcal{E} with infinitely many classes is computable if and only if there is a limitwise monotonic function F (with range $\omega \cup \{\infty\}$) for which there are exactly $|\{x : F(x) = \kappa\}|$ many classes of size κ (for each $\kappa \in \omega \cup \{\infty\}$) in \mathcal{E} .*

Theorem 1.4 can be rephrased to say *the computable isomorphism types of computable equivalence structures are specified by limitwise monotonic functions*.

It turns out that there are a number of applications of limitwise monotonicity in the literature, and we will explore some of these in this paper. They include quite a number of problems in linear orderings, trees, p -groups, computable spectra of \aleph_1 -categorical structures, and general computable model theory in and around

prime models. Many of these applications require that the notion be applied in relativized form, and we will explore this, proving some general theorems about such sets.

Because of the connection between limitwise monotonic sets and applications, we will try to understand when it is possible to find a nonlimitwise monotonic set below a given degree. We therefore introduce the following concept.

Definition 1.5. A degree \mathbf{a} is *totally limitwise monotonic* if every set $B \leq_T \mathbf{a}$ is a limitwise monotonic set.

We prove the following result.

Theorem 1.6. *A computably enumerable degree \mathbf{a} is totally limitwise monotonic if and only if \mathbf{a} is non-high.*

Thus, for example, if a non-high computably enumerable degree \mathbf{a} can compute a set, then there is a computable equivalence structure having classes of exactly those sizes.

Additionally, we will look at the situation where limitwise monotonicity seems not to be enough, but certain variations suffice, such as η -representations. Here we will show that for the question of strong η -representations, the variation of limitwise monotonicity introduced by Kach and Turetsky (see [19]) does not suffice.

Theorem 1.7 (Turetsky). *There is a set S with a computable strong η -representation that is not support strictly increasing $\mathbf{0}'$ -limitwise monotonic on \mathbb{Q} .*

In the last section, we introduce a new notion which we term the limitwise monotonic spectrum of a set. The idea here is that we wish to recast the relationship between limitwise monotonicity and degrees of unsolvability in a more abstract setting. This leads to the following definition.

Definition 1.8. If $S \subseteq \omega$ is any nonempty set, define $\text{LMSpec}(S)$ to be the set

$$\text{LMSpec}(S) := \{\mathbf{a} : S \text{ is } \mathbf{a}\text{-limitwise monotonic}\}.$$

In addition to rephrasing existing results, we show if $\mathbf{a} < \mathbf{b}$, then there exists a set S with $\mathbf{a} \notin \text{LMSpec}(S)$ and $\mathbf{b} \in \text{LMSpec}(S)$. While we do not develop this subject further, we believe this may have wider applications.

2. LIMITWISE MONOTONIC FUNCTIONS AND SETS

As a first step towards understanding the limitwise monotonic sets, it is natural to determine where they sit in the arithmetic hierarchy. It is also important to recognize that being a limitwise monotonic set is not degree invariant.

Theorem 2.1 (Folklore).

- (i) *If A is a limitwise monotonic set, then A is Σ_2^0 .*
- (ii) *If A is Σ_2^0 , then $A \oplus \omega$ is a limitwise monotonic set.*

Proof. (i) Let f be a computable approximation function witnessing that A is a limitwise monotonic set. Then $n \in A$ if and only if $(\exists x)(\exists s)(\forall t \geq s) [f(x, t) = n]$.

(ii) Let A be Σ_2^0 , so that $n \in A$ if and only if $(\exists s)(\forall t) [R(n, s, t)]$. For convenience, we assume $(\forall n) [\neg R(n, 0, 0)]$. Then

$$f(\langle n, s \rangle, t) = \begin{cases} 2n & \text{if } (\forall t' < t) [R(n, s, t')], \\ 2n + 1 & \text{otherwise} \end{cases}$$

witnesses that $A \oplus \omega$ is a limitwise monotonic set. \square

The latter is essentially an idea of Lerman (see [24]). As implicitly noted there, this result generalizes to any nonimmune Σ_2^0 set in place of ω . Indeed, it is not hard to show any Σ_2^0 set containing a limitwise monotonic set is itself a limitwise monotonic set.

It might be tempting to conjecture that every Σ_2^0 set is a limitwise monotonic set, but this is not true. Though every c.e. set is clearly limitwise monotonic, not every d.c.e. set is a limitwise monotonic set. Recall a set A is *d.c.e.* if there are c.e. sets B and C with $A = B - C$.

Theorem 2.2 (Khousainov, Nies, and Shore [22]). *There is a Δ_2^0 set A , indeed a d.c.e. set A , which is not a limitwise monotonic set.*

Proof (Sketch). We construct a d.c.e. set A satisfying the requirements:

\mathcal{R}_e : The function $\varphi_e(\cdot, \cdot)$ does not witness that A is a limitwise monotonic set.

Towards meeting \mathcal{R}_0 , we pick a witness n_0 and put n_0 into A . We then wait for $\varphi_0(x, t_0) = n_0$ for some x and stage $t_0 > n_0$. At such a stage t_0 , we remove all elements between n_0 and t_0 currently in A , and allow only elements n larger than t_0 to enter A . We maintain this \mathcal{R}_0 restraint until a stage $t_1 > t_0$ and number $n_1 > n_0$ is found with $\varphi_0(x, t_1) = n_1$ and $n_1 \in A$. At such a stage t_1 , we put $n_0 + 1$ into A . We then repeat this process by taking all elements between n_1 and t_1 out of A , releasing the previous A restraint, and allowing only elements n less than t_0 or bigger than t_1 to enter A .

If for every i we find a stage t_i and number n_i , then the values of $\varphi_0(x, s)$ tend to infinity, satisfying \mathcal{R}_0 . Lower priority strategies \mathcal{R}_e can then play “behind” this \mathcal{R}_0 restraint. If instead there is an i such that t_i and n_i are never found, then $\lim_s \varphi_0(x, s) \notin A$, satisfying \mathcal{R}_0 . Lower priority strategies can then play “above” this finitary restraint.

The interaction of strategies is straightforward as above, with each strategy guessing whether higher priority strategies have a finitary or infinitary outcome. \square

In the definition of a limitwise monotonic set, we imposed no restraint on the number of times an element could appear in the range of F . One might think that such a restraint would give a stronger notion, but this turns out not to be the case.

Theorem 2.3 (Harris [14]). *If F is a limitwise monotonic function with infinite range, then there is an injective limitwise monotonic function G with $\text{range}(F) = \text{range}(G)$.*

Proof. Fixing a limitwise monotonic approximation f for F , we define a limitwise monotonic approximation g for G . Indeed, we define

$$g(m, s) = \begin{cases} 0 & \text{if } m \geq s, \\ f(n, t) & \text{otherwise, where } \langle n, t \rangle \text{ with } t \geq s \text{ is least so that} \\ & f(n, t) \neq g(x, s) \text{ for all } x < m \text{ and } f(n, t) \geq g(m, s - 1). \end{cases}$$

As g is clearly computable (note n and t must exist as $\text{range}(F)$ is infinite) and nondecreasing, it suffices to argue that $G(m) = \lim_s g(m, s)$ exists for all m , that G is injective, and that $\text{range}(F) = \text{range}(G)$.

Induction demonstrates the limit $G(m)$ exists for all m . Indeed, if $G(0), \dots, G(m - 1)$ all exist, fix a stage after which these limits are achieved. Then either

$g(m, s) < \max\{G(0), \dots, G(m-1)\}$ for all s (in which case $G(m)$ exists) or there exists an $s_0 \in \omega$ with $g(m, s_0) > \max\{G(0), \dots, G(m-1)\}$. In the latter case, the value of n used in defining $g(m, s)$ can only change finitely often, from which it follows that $G(m)$ exists.

The injectivity of G follows from g being injective at every stage, i.e., that $g(m, s) \neq g(m', s)$ if $m \neq m'$.

Finally, we argue $\text{range}(F) = \text{range}(G)$. As $\text{range}(G) \subseteq \text{range}(F)$ is immediate, we demonstrate that $F(n) \in \text{range}(G)$ for all $n \in \omega$ by induction on n . If $F(0), \dots, F(n-1) \in \text{range}(G)$, fix a stage s_0 for which $F(k) = f(k, s)$ for all $s \geq s_0$ and $k \leq n$ and for which $g(m, s) = F(k)$ for all $s \geq s_0$ and m witnessing $F(0), \dots, F(n-1) \in \text{range}(G)$. Then at stage s_0 (provided $s_0 > n$), either $g(x, s) = f(n, s)$ for some $x < s_0$ or $g(s_0, s_0)$ will be defined as $F(n) = f(n, s_0)$. In the former case, either $G(x) = F(n)$ or $G(y) = F(n)$ for some $y < x$; in the latter case, either $G(s_0) = F(n)$ or $G(x) = F(n)$ for some $x < s_0$. Thus $F(n) \in \text{range}(G)$. \square

Because of its use in applications, it is natural to wonder which degrees compute a Σ_2^0 set that is not a limitwise monotonic set. This motivates our notion of a *totally limitwise monotonic degree* (recall Definition 1.5).

Theorem 2.4. *A computably enumerable degree \mathbf{a} is totally limitwise monotonic if and only if \mathbf{a} is non-high.*

Proof (\implies). We show if \mathbf{a} is high, then it computes a nonlimitwise monotonic set. Fixing a c.e. set $A \in \mathbf{a}$ with enumeration $\{A_s\}_{s \in \omega}$, as \mathbf{a} is high there is an A -computable function f^A that is dominating with respect to the class of all computable functions. Denote the use of f^A by g , so that $g(n)$ is the use of the computation $f^A(n)$. We build a set $S \leq_T A$ (we witness this by $S = \Gamma^A$) that is not a limitwise monotonic set by diagonalizing against all candidate approximation functions $\{\varphi_i\}_{i \in \omega}$. We introduce the obvious requirements.

\mathcal{R}_i : The function Φ_i does not witness that S is a limitwise monotonic set.

The strategy to satisfy \mathcal{R}_i is similar to that in Theorem 2.2 except here we need an A -permission to remove a witness n_0 put into S . As this A -permission may never appear, there is a need to choose another witness n_1 and restart this process. We build a computable function h , extending it whenever we need a permission. That f^A is dominating will guarantee that eventually a witness will receive an A -permission.

Strategy for \mathcal{R}_i :

- (1) Choose an integer n_0 and put n_0 in B via $\Gamma^{A_s \upharpoonright g(n_0)}(n_0) = 1$.
- (2) Wait for a column x_0 for which $\varphi_i(x_0, s) = n_0$.
- (3) For $n \leq n_0$, if $h(n)$ is not yet defined, define $h(n) = f^{A_s}(n_0)$.
- (4) Wait for $A_{s+1} \upharpoonright g(n_0) \neq A_s \upharpoonright g(n_0)$. While waiting, return to Step 1, instead using a new integer n_{k+1} .
- (5) If an A -permission is seen, remove n_0 and all later n_k from B via $\Gamma^{A_s \upharpoonright g(n_0)}(n_k) = 0$, and cancel all work for later n_k .
- (6) Keep n_0 and all later n_k out of B , and wait for $\varphi_i(x_0, s)$ to increase to some n_1 currently in B . When this occurs, return to Step 3 with n_1 in place of n_0 .

Of course, if the strategy spends cofinitely many stages in Step 2 with some n_k , then \mathcal{R}_i is satisfied as $n_k \in S$ and $n_k \notin \text{range}(\Phi_i)$. If the strategy spends cofinitely many stages in Step 6 for some n_k , then $\Phi_i(x_j) \notin S$. If the strategy sees $A_{s+1} \upharpoonright g(n_k) \neq A_s \upharpoonright g(n_k)$ for cofinitely many k , then \mathcal{R}_i is satisfied as $\Phi_i(x_j)$ is not defined (having an infinite limit). However, one of these must be the case, as if we wait at Step 4 forever with infinitely many k , then h is not dominated by f^A , a contradiction.

As before, the interaction of strategies is straightforward, with each strategy guessing whether higher priority strategies have a finitary or infinitary outcome. \square

Proof (\Leftarrow). Fixing a nonhigh c.e. degree \mathbf{a} and a c.e. set $A \in \mathbf{a}$ with approximation $\{A_s\}_{s \in \omega}$, we show every $B \leq_T A$ (say $B = \Psi^A$) is a limitwise monotonic set. We approximate B by running Ψ^A with the approximations to A . Let B_s denote our approximation at stage s , i.e., let $B_s = \Psi^A[s]$. We enumerate a subset of B_s recursively as follows:

$$\begin{aligned} b_0^s &= \min B_s \\ b_{n+1}^s &= \min\{b \in B_s \mid (\forall t \leq s)[b > b_n^t]\}. \end{aligned}$$

Before continuing, we argue $\lim_s b_n^s$ exists for all n .

Claim 2.4.1. For every n , the sequence $\{b_n^s\}_{s \in \omega}$ converges to a finite limit.

Proof. Clearly, we have $\min B = \lim_s b_0^s$. Assuming $\lim_s b_n^s$ exists, we show $\lim_s b_{n+1}^s$ exists. As $\lim_s b_n^s$ exists by hypothesis, the set $\{b_n^s\}_{s \in \omega}$ is finite, so we may let $b \in B$ be least such that b is greater than all these values. Let s_0 be a stage by which B has converged on b and b_n^s has converged. Then for any stage $s > s_0$, we have $b_{n+1}^s = b$. \square

With this, we define a total function $f = \Gamma^A$. Using a witness to the failure of f being a dominating function for the class of total computable functions, we demonstrate B is a limitwise monotonic set. As preparation, let $\{\varphi_i\}_{i \in \omega}$ be an effective listing of all partial computable functions for which $\varphi_{i,s}(n) \downarrow$ implies $\varphi_{i,s}(m) \downarrow$ for all $m < n$.

Initially, we define $\Gamma^{A_0}(n) = 0$ for all n with use $A_0 \upharpoonright \psi(b_n^0)$. At stage $s + 1$, if $\Gamma^{A_s}(n)$ is undefined, we define it by

$$\Gamma^{A_s}(n) = \max\{\varphi_{i,s}(n) : i < n \text{ and } \varphi_{i,s}(n) \downarrow\} + 1$$

with use $A_s \upharpoonright \psi(b_n^s)$.

As a consequence of Claim 2.4.1, it follows that $\psi(b_n^s)$ converges. Thus f is a total A -computable function. As \mathbf{a} is nonhigh, we may fix an index k such that φ_k is a total computable function which f does not dominate. From this, as already suggested, we construct a limitwise monotonic approximation h to B .

Construction: At stage $s = 0$, we define $h(x, 0) = 0$ for all x .

At stage $s + 1$, if φ_k has not converged on any new values since stage s , we define $h(x, s + 1) = h(x, s)$ for all x . If $\varphi_k(n)$ newly converges at stage s for some $n \leq s$, then for every $b \in B_s$ with $b \leq b_n^s$, we choose a previously unused x and define $h(x, s + 1) = b$. For every x with $h(x, s) \notin B_s$, we define $h(x, s + 1) = b_m^s$ for the least $m \leq n$ with $b_m^s > h(x, s)$. For all other $x \leq s$, we define $h(x, s + 1) = h(x, s)$.

Verification: We verify that h is a limitwise monotonic approximation to B . By construction, the function h is total and computable. Define $H(x) = \lim_s h(x, s)$. We verify that $H(x)$ exists and that H witnesses that B is a limitwise monotonic set.

Claim 2.4.2. The function H is total.

Proof. Fixing an integer x , we may suppose there is a stage s_0 with $h(x, s_0) > 0$ (else $H(x) = 0$). Let n be an integer such that $\varphi_k(n)$ has not converged by stage s_0 , and $\varphi_k(n) \geq f(n)$. Let $s_1 > s_0$ be the stage at which $\varphi_k(n)$ converges. Then b_n^s has necessarily converged by stage s_1 . By our assumption that $\varphi_k(n)$ converges before $\varphi_k(m)$ for any $m > n$, we have $h(x, s_1) \leq b_n^{s_1}$. Thus $H(x) \leq b_n^{s_1}$. \square

Claim 2.4.3. The function H enumerates B .

Proof. For any $b \in B$, choose a stage s_0 such that B_s has converged on b , and an n such that $b_n^{s_0} > b$. When φ_k converges on any $m > n$, an x will be created such that $h(x, s) = b$, and this will never change at later s . Thus $H(x) = b$.

For any $c \notin B$, choose a stage s_0 such that B_s has converged on c , and an n such that $b_n^{s_0} > c$. When φ_k converges on any $m > n$, any x with $h(x, s) = c$ will change their value. Thus, for no x does $H(x) = c$. \square

This completes the proof. \square

The hypothesis that \mathbf{a} is c.e. in Theorem 2.4 is very much necessary as the following result shows.

Theorem 2.5 (Hirschfeldt, R. Miller, and Podzorov [15]). *There is a low Δ_2^0 set A which is not a limitwise monotonic set.*

Proof (Sketch). In addition to the \mathcal{R}_e requirements of Theorem 2.2, we meet the standard lowness requirements:

$$\mathcal{N}_e : \exists^\infty s [\Psi_e^A(e)[s] \downarrow] \implies \Psi_e^A(e) \downarrow.$$

To allow \mathcal{N}_e to be met in the presence of the \mathcal{R}_e requirements of Theorem 2.2, we use the fact that A is Δ_2^0 . More specifically, we allow \mathcal{R}_j for $j < e$ to injure \mathcal{N}_e . This would seem problematical since the action of \mathcal{R}_j may be infinitary. However, this is not the case.

For example, consider a single higher priority \mathcal{R}_j requirement. If we are at a stage where we see some computation $\Psi_e^A(e)[s] \downarrow$ and the use $\psi_e^A(e)[s]$ is less than the number n_i currently being used for \mathcal{R}_j , then \mathcal{N}_e can assert control of $A \upharpoonright \psi_e^A(e)[s]$ and preserve the computation with impunity. On the other hand, it might be that \mathcal{R}_j is pointing at some n_i below the use $\psi_e^A(e)[s]$. In this case, what \mathcal{N}_e will do is assert control and restrain this portion of A with priority e . Of course \mathcal{R}_j can later injure this, but when it does it must move to a new n_{i+1} and this will be large. It might happen that at some stage $s' > s$, again \mathcal{N}_e might try to preserve some new computation $\Psi_e^A(e) \downarrow [s']$ and this new computation can be injured by \mathcal{R}_j again. *But* once we pick yet another n_{i+2} , we note that \mathcal{R}_j will only be concerned with numbers *bigger* than $n_{i+1} > \psi_e^A(e)[s]$. With no injury to \mathcal{R}_j we can therefore *restore* the stage $A \upharpoonright (\psi_e^A(e)[s])[s'] = A \upharpoonright \psi_e^A(e)[s]$ and then this computation $\Psi^A(e)[s'] = \Psi^A(e)[s]$ with no possible future injury from \mathcal{R}_j .

In this way, with a finite injury argument we can meet all the \mathcal{N}_e . \square

The final result in this section is a basis theorem for Π_1^0 trees. Its proof is not illuminating, so we omit it.

Theorem 2.6 (J. Miller [19]). *If $P \subseteq 2^\omega$ is a nonempty Π_1^0 class containing a nonempty set, then P contains a limitwise monotonic set.*

Thus, for example, there are limitwise monotonic sets that are 1-random and that are DNR_2 .

3. APPLICATIONS OF LIMITWISE MONOTONIC FUNCTIONS AND SETS

Though equivalence structures appear to be the simplest application of limitwise monotonic sets to effective algebra, historically limitwise monotonic functions were introduced by Khisamiev as a means of characterizing the computable reduced abelian p -groups of length ω (see [20]). There, these functions were termed *s-functions*.

Theorem 3.1 (Khisamiev [20]). *A reduced abelian p -group \mathcal{G} of length ω is computable if and only if there is a limitwise monotonic function F such that $u_n(\mathcal{G}) = |\{x : F(x) = n\}|$.*

Though the proof of Theorem 3.1 is more involved than Theorem 1.4, the idea is very much the same. The height of an element x in the group $\mathcal{G}[s]$ is computable from a computable presentation, and this height can only increase as s increases. Conversely, it is easy to build \mathcal{G} from a computable approximation f to the limitwise monotonic function F .

After Khisamiev's work, other applications were discovered in effective algebra, particularly within the context of linear orders. The simplest (infinite) orderings are presentations of ω . The simplest relation we could study on such orderings would be a unary relation. This began with Downey, Khoussainov, J. Miller, and Yu (see [11]) (which was circulated for a long time in preprint form), then Hirschfeldt, R. Miller, and Podzorov (see [15]), and finally Knoll (see [23]).

Definition 3.2 (Hirschfeldt, R. Miller, and Podzorov [15]). A set A is *order computable* if there is a computable copy of $(\omega : <, A)$ in the language of linear orderings with an additional unary predicate.

Observation 3.3 (Kach and Turetsky [19]). *Every order computable set is a limitwise monotonic set.*

The proof of this observation is immediate. When we see some element n declared to be in the set representing A in the computable copy of $(\omega : <, A)$, then n can only move to bigger things and has a limit. Since every c.e. set is limitwise monotonic, the following result says that order computability is a significantly more refined concept than limitwise monotonicity.

Theorem 3.4 (Downey, Khoussainov, J. Miller, and Yu [11]). *Every high c.e. degree contains a c.e. set which is not order computable.*

The last application we give of (unrelativized) limitwise monotonicity is within computable model theory. Baldwin and Lachlan showed that for an uncountably categorical but not countably categorical theory T , the countable models form an elementary chain of length $\omega + 1$. An interesting line of research has been determining whether, for a fixed set $S \subseteq \omega + 1$, there is a theory T whose computable

models (identifying a model with its position in the elementary chain) are precisely those in S . Khoussainov, Nies, and Shore realized the subset $(\omega + 1) - \{0\}$ via limitwise monotonic sets.

Theorem 3.5 (Khoussainov, Nies, and Shore [22]). *There is an uncountably categorical but not countably categorical theory T for which every model but the prime model is computable (realizing the set $(\omega + 1) - \{0\}$).*

Proof (Sketch). The desired theory T is in the language of infinitely many binary predicates $\{P_i\}_{i \in \omega}$. For each $n \in \omega$, an n -cube is a collection of 2^n many elements isomorphic to the structure with universe $\{x_\tau\}_{\tau \in 2^n}$ satisfying $P_i(x_\tau, x_{\tau'})$ if and only if $x_\tau(i) \neq x_{\tau'}(i)$ and $x_\tau(j) = x_{\tau'}(j)$ for all $j < i$. An ω -cube is the union of a chain of n -cubes for all $n \in \omega$.

From a computable structure consisting only of finite n -cubes, it is possible to approximate from below the maximal integer n to which a given element is part of an n -cube. Thus any set $S \subseteq \omega$ is a limitwise monotonic set if the structure consisting of (exactly) one n -cube for each $n \in S$ is computable.

Conversely, given an approximation of an integer n from below, it is possible to uniformly construct a computable presentation of an n -cube. Thus if S is a limitwise monotonic set, the structure consisting of (exactly) one n -cube for each $n \in S$ is computable.

It therefore suffices to fix a Σ_2^0 set S that is not a limitwise monotonic set. Let T be the theory of the model containing (exactly) one n -cube for each $n \in S$. Any non-prime model contains an ω -cube, and is thus computable as any wrongly constructed n -cubes can be grown into a fixed ω -cube. On the other hand, the prime model cannot be computable as S was chosen not limitwise monotonic. \square

4. RELATIVIZED LIMITWISE MONOTONICITY

For many other applications of limitwise monotonicity in effective algebra and computable model theory, the notion needs to be relativized. In many of these cases, a relativized version of Theorem 2.2 is useful.

Definition 4.1. A function F is **a**-limitwise monotonic if there is an **a**-computable approximation function $f(\cdot, \cdot)$ such that, for all x ,

- (i) $F(x) = \lim_s f(x, s)$.
- (ii) For all s , $f(x, s) \leq f(x, s + 1)$.

A set S is **a**-limitwise monotonic if it is the range of a **a**-limitwise monotonic function.

Corollary 4.2. *There is a Δ_3^0 set A which is not $\mathbf{0}'$ -limitwise monotonic.*

Perhaps the most direct application of $\mathbf{0}'$ -limitwise monotonicity is to linear orderings. In the same spirit as ζ -representations (see Definition 1.1), we can define the *strong η -representation* of an infinite set S and the *weak η -representation* by replacing the order type ζ with the order type η . It is easy to see that the set of maximal block sizes is a $\mathbf{0}'$ -limitwise monotonic set if the η -representation is computable, as $\mathbf{0}'$ can decide if a pair of points form an adjacency. Partially answering a question of Downey (see [10]), Harris established the following.

Theorem 4.3 (Harris [14]). *A set S has a computable weak η -representation if and only if S is $\mathbf{0}'$ -limitwise monotonic.*

We will sketch the proof of Theorem 4.3. We begin with a result independently proven by Harris and Kach.

Proposition 4.4 (Harris [14] and Kach [17]). *A function F is \mathbf{O}' -limitwise monotonic if and only if there is a computable function $g(\cdot, \cdot)$ such that $F(n) = \liminf_s g(n, s)$.*

Kach defined a set A to be a *limit infimum set* if there is a computable function g as in Proposition 4.4 with A the range of $\liminf_s g(\cdot, s)$. Then Proposition 4.4 can be rephrased as *a set A is a limit infimum set if and only if A is a \mathbf{O}' -limitwise monotonic set*.

Proof of Proposition 4.4. Let f witness that F is \mathbf{O}' -limitwise monotonic. By the Limit Lemma, there is a computable function h such that $f(n, s) = \lim_t h(n, s, t)$ for all n and s . Fixing n , the idea is to view $h(n, 0, t), h(n, 1, t), \dots, h(n, t, t)$ as approximations to $F(n)$. We define $g(n, t)$ to be the maximum value of $h(n, j, t)$ for $j < i$, where i is maximal so that $h(n, i, t) = h(n, i, t - 1)$. It can be verified that g is indeed a limit infimum approximation to F with the property $F(n) = G(n) := \liminf_t g(n, t)$.

For the reverse direction, let g witness that F is limit infimum. Defining $f(n, s)$ by $f(n, s) = \min\{g(n, t) : t \geq s\}$ yields an approximation function that is readily verified to be a \mathbf{O}' -limitwise monotonic approximation to F with the property $F(n) = \lim_s f(n, s)$ where $F(n) := \liminf_t g(n, t)$. \square

Proof of Theorem 4.3. Via Proposition 4.4, we may fix a function g witnessing that S is limit infimum. Then, for each n , we will build a block of size $\liminf_s g(n, s)$ in stages, putting a dense ordering between the blocks. The action at stage $s + 1$ depends on the relative sizes of $g(n, s)$ and $g(n, s + 1)$: if $g(n, s + 1) > g(n, s)$, we can add to the *outside* of the current $g(n, s)$ block so that it has size $g(n, s + 1)$; and if $g(n, s + 1) < g(n, s)$, we can remove the *outside* points and incorporate them into the adjacent interval of order type η . Independent of the relative sizes, we work towards making the endpoints limit points with the order type η between adjacent blocks. \square

Analyzing the proof a bit more carefully, it is easy to see that the maximal blocks created appear in the same order (and thus with the same multiplicity) as given by F . Using Theorem 2.3, Harris was thus able to characterize the sets S with a computable *unique η -representation*, i.e., any order type where the function f in Definition 1.1 (with the order type η replacing the order type ζ) is injective.

Corollary 4.5 (Harris [14]). *A set S has a computable unique η -representation if and only if S is \mathbf{O}' -limitwise monotonic.*

Another application of \mathbf{O}' -limitwise monotonicity in linear orderings concerns shuffle sums. Recall that the *shuffle sum* of a set S is a linear ordering obtained by taking the rationals and replacing each element by a block of cardinality a member of S in such a way that the blocks representing the members of S occur densely.

Theorem 4.6 (Kach [17]). *The shuffle sum of a set S is computable if and only if S is \mathbf{O}' -limitwise monotonic.*

Proof (Sketch). Via Proposition 4.4, again fix a function g so that S is the range of $\liminf_s g(\cdot, s)$. This time we build blocks of size $g(n, s)$ densely at every stage. Again, we increase or cut back the size of blocks depending on whether $g(n, s)$

increases or decreases. The only complication is points which are removed cannot be so easily incorporated as in Theorem 4.3. Instead, new blocks will recycle these rejected points based on a priority ranking. \square

Part of the interest for understanding which shuffle sums are computable stems from an earlier result in computable model theory by Hirschfeldt, answering a question of Rosenstein (see [29]).

Theorem 4.7 (Hirschfeldt [16]). *There is a complete theory T in the language of linear orders having a prime model and a computable model, but no computable prime model.*

Proof. By Theorem 2.2, fix a set $S \in \Sigma_3^0$ that is not \mathbf{O}' -limitwise monotonic. Let T be the theory of the shuffle sum of the set S . Then the shuffle sum of S is the prime model; however it is not computable as S was not \mathbf{O}' -limitwise monotonic. On the other hand, the shuffle sum of S with ζ is computable and is a model of T . \square

The earliest application of \mathbf{O}' -limitwise monotonicity to linear orderings seems to concern isomorphism types of initial segments (i.e., convex initial sets) of linear orderings. Initial segments of linear orderings can be extremely complex, as witnessed, for example, by the initial segment of order type ω_1^{CK} of the Harrison ordering $\omega_1^{\text{CK}}(1 + \eta)$. Of interest here is how complex the classical isomorphism type of an initial segment of a computable ordering can be to still be assured a computable presentation. For example, Rosenstein asked whether every Π_2^0 initial segment of a computable linear ordering is isomorphic to a computable linear ordering (see [29]). Rosenstein had already demonstrated that this could not be strengthened to Π_3^0 using index sets (like Feiner's Theorem, see [12]). From the other direction, Raw showed every Σ_1^0 initial segment of a computable linear ordering is isomorphic to a computable linear ordering (see [28]). This was improved by Ambos-Spies, Cooper, and Lempp to every Σ_2^0 initial segment of a computable linear ordering has a computable copy (see [1]). Coles, Downey, and Khoussainov closed the gap, answering Rosenstein's question, by exhibiting a computable linear ordering with a Π_2^0 initial segment not isomorphic to a computable linear ordering. The initial segment is an η -like linear order, i.e., the result of replacing every element of the rationals with a finite block.

Theorem 4.8 (Coles, Downey, and Khoussainov [9]). *For any set $S \subseteq \omega$, then $S \in \Sigma_3^0$ if and only if there is a computable linear ordering \mathcal{L} of the form $\mathcal{L} = \mathcal{A} + \mathcal{B}$ with \mathcal{A} an η -like linear order having maximal blocks of sizes exactly those numbers in S and $\mathcal{B} \cong \omega^*$.*

The proof of Theorem 4.8 requires a reasonably nontrivial Π_2^0 priority argument (see [9] for details). But, granted Theorem 4.8, we can deduce the following.

Theorem 4.9 (Coles, Downey and Khoussainov [9]). *There is a computable linear ordering with a Π_2^0 initial segment not isomorphic to a computable linear ordering.*

Proof. By Corollary 4.2, there is a Σ_3^0 set S which is not a \mathbf{O}' -limitwise monotonic set. By Theorem 4.8, there is a computable linear ordering $\mathcal{L} = \mathcal{A} + \mathcal{B}$. Then \mathcal{A} is not computable (as an isomorphism type) as S was not \mathbf{O}' -limitwise monotonic. However, the set A is Π_2^0 , being the set of points having infinitely many points to the right. \square

The last application within the context of linear orderings involves the complexity of subsets rather than the complexity of intervals. Kach and J. Miller used relativizations of limitwise monotonic functions to each degree $\mathbf{0}^{(n)}$ for $n \in \omega$ to prove the following result.

Theorem 4.10 (Kach and J. Miller [18]). *There is a computable non-well-ordered intrinsically computably well-ordered linear order, i.e., there is a computable non-well-ordered linear order for which no computable presentation has a computable subset of order type ω^* .*

Though the proof of Theorem 4.10 is rather involved, the major idea is that $\mathbf{0}^{(2n+1)}$ can approximate the value of $F(n)$ in the linear ordering $\cdots + \omega^n \cdot F(n) + \cdots + \omega^2 \cdot F(2) + \omega \cdot F(1) + F(0)$ in a monotonic manner.

Despite all the discussed applications of relativized limitwise monotonic functions being in the context of linear orderings, Khisamiev first relativized limitwise monotonic functions in the context of reduced abelian p -groups.

Theorem 4.11 (Khisamiev [21]). *A reduced abelian p -group \mathcal{G} of length less than ω^2 (say of length at most $\omega \cdot N$) is computable if and only if there are functions F_0, F_1, \dots, F_{N-1} such that F_i is $\mathbf{0}^{(2i)}$ -limitwise monotonic and $u_{\omega \cdot i + n}(\mathcal{G}) = |\{x : F_i(x) = n\}|$.*

Proof (Sketch). If \mathcal{G} has length less than ω^2 , (nonuniformly) fix elements g_1, g_2, \dots, g_N with the height of g_i being $\omega \cdot i$. The oracle $\mathbf{0}^{(2i)}$ is powerful enough to approximate the height of elements below g_i , yielding a limitwise monotonic approximation function. Appealing to Theorem 2.1, it follows (uniformly, though such uniformity is unnecessary) there is such a sequence of function F_i . \square

5. BEYOND LIMITWISE MONOTONICITY

As with order computable sets, sometimes (relativized) limitwise monotonicity fails to fully capture some algebraic phenomenon. A recent example of this was demonstrated by Kach and Turetsky in their work on an old question going back to Rosenstein (see [29]) and Downey (see [10]). Generalizing the notion of a strong η -representation, an *increasing η -representation* of S is a linear order \mathcal{L} of the form $\eta + n_0 + \eta + n_1 + \dots$ where the n_i enumerate S in increasing order (possibly with repeats). It is not hard to show that not all $\mathbf{0}'$ -limitwise monotonic sets S have non-decreasing η -representations since such an S needs to be Δ_3^0 .

To analyze the question of what $\mathbf{0}'$ -limitwise monotonic sets S have strong or increasing η -representations, we introduce a new class of sets.

Definition 5.1 (Kach and Turetsky [19]). A function $F : \mathbb{Q} \rightarrow \omega$ is *support (strictly) increasing* if $F(q_1) \leq F(q_2)$ ($F(q_1) < F(q_2)$) whenever $q_1 < q_2$ and $F(q_1), F(q_2) > 0$, the range of F is unbounded, and the support of F has order type ω .

A function $F : \mathbb{Q} \rightarrow \omega$ is *support (strictly) increasing limitwise monotonic on \mathbb{Q}* if it is support (strictly) increasing and there is a computable approximation function $f : \mathbb{Q} \times \omega \rightarrow \omega$ such that $F(q) = \lim_s f(q, s)$ and $f(q, s) \leq f(q, s+1)$.

The intuition here is that most $F(q)$ will be zero, but once we see $F(q) > 0$ at some stage (when $f(q, s) > 0$), then we “know” its relationship with all those q' with $F(q') > 0$.

Theorem 5.2 (Kach and Turetsky [19]). *A set S has a computable increasing η -representation (with only finitely many blocks of any size $n > 1$) if and only if S is support increasing \mathbf{O}' -limitwise monotonic on \mathbb{Q} .*

Proof (Sketch). The forward direction is clear since given a \mathbf{O}' -oracle we know the blocks (monotonically) and how to order them. Of course, within the construction we might think that we have (distinct) blocks around q_1 and q_2 . At a later stage, we may see these blocks merge, causing us to have an “extra” column of F which is positive. This is easily remedied by having $h(q_1, s) = h(q_2, s)$ reflect the merged size for all later s .

For the reverse direction, the construction proceeds as in Theorem 4.3. The only difference is that whenever we see $F(q) > 0$ for some new q (i.e., we see $f(q, s) > 0$), we create a new finite block within the linear order at the appropriate place. \square

It would be nice if altering the domain to \mathbb{Q} would have application in characterizing the sets with computable strong η -representations. Frolov and Zubkov (see [13]) and Kach and Turetsky (see [19]) have shown that there is a support increasing \mathbf{O}' -limitwise monotonic set on \mathbb{Q} not having a computable strong η -representation. We finish this section by showing that being support strictly increasing \mathbf{O}' -limitwise monotonic on \mathbb{Q} is not necessary to have a computable strong η -representation.

Theorem 5.3 (Turetsky). *There is a set S with a computable strong η -representation that is not support strictly increasing \mathbf{O}' -limitwise monotonic on \mathbb{Q} .*

Proof. Let $\{f_i(x, s)\}_{i \in \omega}$ be an enumeration of candidate total \mathbf{O}' -computable monotonic approximations on \mathbb{Q} . By the Limit Lemma, let $\{\hat{f}_i(x, s, t)\}_{i \in \omega}$ be an enumeration of computable approximations to f_i so that $f_i(x, s) = \lim_t \hat{f}_i(x, s, t)$. Note that since the f_i are total, the limit $\lim_t \hat{f}_i(x, s, t)$ will always converge.

We construct a computable presentation of a strong η -representation and let S be the set represented. We meet the following requirements:

\mathcal{R}_i : The set S is not the range of F_i .

The strategy to assure \mathcal{R}_i hinges on the fact that support strictly increasing limitwise monotonic functions cannot cope with two blocks in a strong η -representation merging. This fact is exploited to force a column to infinity.

Strategy for \mathcal{R}_i : Let $<_{\mathbb{Q}}$ be the natural ordering on \mathbb{Q} .

- (1) Choose a large number n_0 and create blocks B_0 and B of sizes $n_0 - 1$ and n_0 in \mathcal{L} at an appropriate location. Restrain other strategies from changing these blocks.
- (2) Wait for a (least) pair $\langle x, u_0 \rangle$ to appear with $\hat{f}_i(x, u_0, t) = n_0$.
- (3) Wait for a (least) pair $\langle x_0, s_0 \rangle$ to appear with $\hat{f}_i(x_0, s_0, t) = n_0 - 1$ and $x_0 <_{\mathbb{Q}} x$.
- (4) Merge B_0 and B and any existing larger blocks into a single block of some size m_0 and release any restraint on this block. Restrain any blocks from forming of sizes between $n_0 - 1$ and m_0 .
- (5) Wait for an $s'_0 > s_0$ with $\hat{f}_i(x_0, s'_0, t) = m'_0$ for some $m'_0 \geq m_0$. If more than one such s'_0 exist, choose the least.
- (6) Release the restraint created at Step 4.
- (7) Wait for a $u_1 > u_0$ with $\hat{f}_i(x, u_1, t) = n_1$ for some $n_1 > m_0$ with n_1 the size of a block in \mathcal{L} .

- (8) Create a block B_1 of size $n_1 - 1$ and restrain other strategies from changing this block or the block found in the previous step. Return to Step 3 with n_1 instead of n_0 .

Note that our actions in Step 4 and Step 8 can be undone — we can resume densifying the interval between B_0 and B to separate the blocks, and we can densify the block B_1 to destroy it. Indeed, this capacity is essential, since there will be times we will need to roll back the construction to an earlier point. If, on some pair we chose, \hat{f}_i changes its value, we return to the step at which we chose it, undoing all work done in the interim.

Thus, if at some stage t , $\hat{f}_i(x, u_0, t) \neq n_0$, we roll back the construction to Step 2. If at some stage t , $\hat{f}_i(x_j, s_j, t) \neq n_j - 1$, we roll back the construction to Step 3 in the j th loop. If at some stage t , $\hat{f}_i(x_j, s'_j, t) \neq m'_j$, we roll back the construction to Step 5 in the j th loop, reestablishing the appropriate restraint. If at some stage t , $\hat{f}_i(x, u_j, t) \neq n_j$ (for $j > 0$), we roll back the construction to Step 7 in the j th loop.

Outcomes for \mathcal{R}_i : There are several possible outcomes for the strategy:

- 2:** The strategy is infinitely often at Step 2, either because it waits at this step forever, or because it is infinitely often rolled back to this step. In either case, n_0 does not appear in the range of F_i but does appear as a block size in \mathcal{L} , and thus F_i does not enumerate S .
- $\langle 3, j \rangle$:** The strategy is infinitely often at Step 3 in the j th loop, either because it waits at this step forever, or because it is infinitely often rolled back to this step. Further, none of outcomes **2**, **$\langle 3, j' \rangle$** , **$\langle 5, j' \rangle$** or **$\langle 7, j' \rangle$** with $j' < j$ apply. In this case, $n_j - 1$ does not appear in the range of F_i but does appear as a block size in \mathcal{L} , and thus F_i does not enumerate S .
- $\langle 5, j \rangle$:** The strategy is infinitely often at Step 5 in the j th loop, either because it waits at this step forever, or because it is infinitely often rolled back to this step. Further, none of outcomes **2**, **$\langle 3, j' \rangle$** with $j' \leq j$, or **$\langle 5, j' \rangle$** or **$\langle 7, j' \rangle$** with $j' < j$ apply. In this case, if $F_i(x_j)$ converges, then $F_i(x_j)$ is between $n_j - 1$ and m_j . However, S will have no element between $n_j - 1$ and m_j , and thus F_i does not enumerate S .
- $\langle 7, j \rangle$:** The strategy is infinitely often at Step 7 in the j th loop, either because it waits at this step forever, or because it is infinitely often rolled back to this step. Further, none of outcomes **2**, **$\langle 3, j' \rangle$** or **$\langle 5, j' \rangle$** with $j' \leq j$, or **$\langle 7, j' \rangle$** with $j' < j$ apply. Then if $F_i(x)$ converges, it does so to a value not contained in S . Thus F_i does not enumerate S .
- ∞ : The strategy spends only finitely many stages at every step in every loop. Since $F_i(x) \geq n_j$ for all j , and $n_j < m_j < n_{j+1}$, $F_i(x)$ diverges.

The Tree: We order the outcomes of a strategy by:

$$\infty > \dots > \langle 7, 1 \rangle > \langle 5, 1 \rangle > \langle 3, 1 \rangle > \langle 7, 0 \rangle > \langle 5, 0 \rangle > \langle 3, 0 \rangle > \mathbf{2}$$

As usual for infinite injury arguments, the *true outcome* of a strategy is the limit infimum of the outcomes.

We arrange the strategies on a tree in the usual fashion. When a strategy τ is rolled back, we also roll back the work done by any strategies ρ directly below τ .

If strategy ρ is below some finite outcome of strategy τ , the strategy ρ chooses a large n_0 and works with values larger than those used by τ . It is possible that ρ

will be injured by a later merge step of τ . However, if we return to ρ , it will mean we have rolled back τ to before the merger, thus healing the injury to ρ .

If strategy ρ is below the infinite outcome of strategy τ , the strategy ρ waits for the restraint of τ to move to a sufficiently late interval that there is sufficient room for ρ to work with values beneath the restraint. It chooses its n_0 smaller than the restraint of τ , but larger than the current size of any blocks which existed when ρ was initialized. When ρ wishes to perform a merger, it waits until τ reaches a Step 6. It then performs the merger as described, including merging larger blocks that τ previously used. If at some later point τ is rolled back, the strategy ρ is rolled back with it.

If ρ is below the infinite outcome of τ , it is possible that τ will violate the restraint of ρ (if τ 's n_j is ρ 's m_k). In this case, ρ waits until τ performs a merger, and then reassigns m_k to the value of this new block (so ρ 's m_k is τ 's m_j). Barring roll back, τ will never again violate this restraint.

In this fashion, strategies respect the restraints imposed by strategies directly above them in the tree. Strategies pay no attention to restraints of any other strategies.

Verification: Define the True Path inductively using the limit infimum of the temporary outcomes.

Claim 5.3.1. If τ is along the True Path, and τ is active at stage t and has a restraint at stage t , then that restraint is not currently violated by some ρ directly below τ .

Proof. If ρ is below some finite outcome of τ , it creates blocks of size larger than the restraint of τ . If ρ is below the infinite outcome of τ , it respects the restraint of τ as discussed above. \square

Claim 5.3.2. If τ is along the True Path, and τ is active at stage t and has a restraint at stage t , then that restraint is not currently violated by some ρ off the True Path.

Proof. Note that the restraint is not violated at the stage it is originally imposed.

Assume ρ is not directly below τ , as that case is handled above.

If the True Path follows a finite outcome at the first place it and ρ differ, and ρ is to the left of the True Path, then any activity by ρ between the stage at which the restraint is imposed and the current stage has been rolled back.

If the True Path follows a finite outcome at the first place it and ρ differ, and ρ is to the right of the True path, then ρ cannot act between the stage at which the restraint is imposed and the current stage (as in order for it to act, τ would have to be rolled back, removing the restraint).

If the True Path follows an infinite outcome at the first place it and ρ differ, then let σ be the meet of τ and ρ . Then ρ created blocks above the restraint of σ , while τ imposes its restraint beneath that of σ . \square

Claim 5.3.3. If τ is along the True Path, and τ imposes a restraint, there will come a stage t when either τ will be rolled back to before it imposed this restraint, τ will release this restraint and this release will never be rolled back, or the restraint will never be violated after stage t .

Proof. Suppose that the restraint is neither rolled back nor released by τ . Then τ will wait until the σ above it stop violating the restraint. The strategy σ can only violate the restraint of τ if τ extends the infinite outcome of σ , and if σ has infinite final outcome, it can only be rolled back to any given step finitely many times. Thus, eventually, σ will never again violate the restraint of τ . Since no other strategies are capable of violating the restraint of τ , the restraint is never again violated. \square

Claim 5.3.4. For any block created in \mathcal{L} , the limit infimum of its size is finite.

Proof. Let B be some block created by some strategy τ .

Suppose ρ is some other strategy. Let σ be ρ meet τ . In order for ρ to affect B , either ρ is σ or ρ is below the infinite outcome of σ , and either τ is σ or τ is below the finite outcome of σ . But by our construction of how strategies below an infinite outcome behave, ρ must have been initialized before B was created.

Thus there are only finitely many ρ that can affect B . Further, barring roll back, each strategy will only affect a given block finitely many times. Thus either one of these strategies is infinitely often rolled back, in which case B is constantly returned to a given finite size, or the size of B stabilizes. \square

Claim 5.3.5. There are blocks of arbitrarily large size in \mathcal{L} .

Proof. Let τ be a strategy along the True Path being initialized at stage t such that this initialization will never be rolled back. During initialization, τ creates a large block. Since τ will never have its initialization rolled back, this block will never be destroyed. It may be grown into a larger block, but by the above, some large block will result. Thus \mathcal{L} has arbitrarily large blocks. \square

Claim 5.3.6. Each strategy along the True Path meets its requirement.

Proof. Immediate from construction. \square

This completes the proof. \square

6. LIMITWISE MONOTONIC SPECTRA

As seemingly all of the effective algebra results with limitwise monotonicity relativize, there is a connection between the degree spectra of a structure

$$\text{DegSpec}(\mathcal{S}) := \{\mathbf{a} : \mathcal{S} \text{ is } \mathbf{a}\text{-computable}\}$$

and the limitwise monotonic spectra of a set.

Definition 6.1. If $S \subseteq \omega$ is any nonempty set, define $\text{LMSpec}(S)$ to be the set

$$\text{LMSpec}(S) := \{\mathbf{a} : S \text{ is } \mathbf{a}\text{-limitwise monotonic}\}.$$

In this language, we can reinterpret some of the material in the preceding sections. Since every Σ_2^0 degree has a limitwise monotonic set, we have the following.

Proposition 6.2 (Folklore). *If \mathbf{a} is Σ_2^0 , then there is an $S \in \mathbf{a}$ with $\mathbf{0} \in \text{LMSpec}(S)$.*

Theorem 6.3 (Khousainov, Nies, and Shore [22]). *There exists a Δ_2^0 set S , indeed a d.c.e. set S , with $\mathbf{0} \notin \text{LMSpec}(S)$.*

Corollary 6.4 (Hirschfeldt, R. Miller, and Podzorov [15]). *There exists a low Δ_2^0 set S with $\mathbf{0} \notin \text{LMSpec}(S)$.*

We also demonstrate some new results.

Proposition 6.5. *There is a set S and a minimal pair of degrees \mathbf{a} and \mathbf{b} with $\mathbf{a}, \mathbf{b} \in \text{LMSpec}(S)$ and $\mathbf{0} \notin \text{LMSpec}(S)$.*

Proof. It suffices to fix a minimal pair of high degrees \mathbf{a} and \mathbf{b} . Then $\mathbf{a}, \mathbf{b} \in \text{LMSpec}(\emptyset''' \oplus \omega)$ by Theorem 2.1 (relativized) as $\emptyset''' \in \Sigma_3^0$ and $\Sigma_2^0(\mathbf{a}) = \Sigma_3^0 = \Sigma_2^0(\mathbf{b})$. On the other hand, it must be the case that $\mathbf{0} \notin \text{LMSpec}(\emptyset''' \oplus \omega)$ as $\emptyset''' \oplus \omega \notin \Sigma_2^0$. \square

Proposition 6.6 (Zubkov). *There are sets S and T with $\mathbf{0} \in \text{LMSpec}(S)$, $\text{LMSpec}(T)$ and $\mathbf{0} \notin \text{LMSpec}(S \cap T)$.*

Proof. By Theorem 2.2, fix a set S that is Σ_2^0 but not limitwise monotonic. Then $S \oplus \omega \oplus \emptyset$ and $S \oplus \emptyset \oplus \omega$ are limitwise monotonic but their intersection is not. \square

Proposition 6.7. *The containment $\text{LMSpec}(T) \subseteq \text{LMSpec}(S)$ does not follow from $S \leq_T T$.*

Proof. By Theorem 2.2, there is a Δ_2^0 set that is not a $\mathbf{0}$ -limitwise monotonic set, yet \emptyset' is a $\mathbf{0}$ -limitwise monotonic set. \square

Theorem 6.8. *If \mathbf{a} and \mathbf{b} satisfy $\mathbf{a} < \mathbf{b}$, then there is a set S with $\mathbf{b} \in \text{LMSpec}(S)$ and $\mathbf{a} \notin \text{LMSpec}(S)$.*

Proof. We start by noting that we may restrict attention to the case when $\mathbf{b} \in \Delta_2^0(\mathbf{a})$. For if $\mathbf{b} \notin \Delta_2^0(\mathbf{a})$, then (with $B \in \mathbf{b}$) either the set B or \overline{B} suffices. The reason is both are clearly \mathbf{b} -limitwise monotonic. If both are \mathbf{a} -limitwise monotonic, then both are $\Sigma_2^0(\mathbf{a})$ by Theorem 2.1. Being complements of each other, this implies both are $\Delta_2^0(\mathbf{a})$, contrary to the hypothesis.

We treat the case when $\mathbf{a} = \mathbf{0}$ and $\mathbf{b} \in \Delta_2^0$ ($\mathbf{b} \neq \mathbf{0}$), with the more general case following by relativization. The idea is to construct a \mathbf{b} -computable limitwise monotonic approximation function $f(x, s)$ such that $\text{range}(F)$ is not the range of any limitwise monotonic function $\{\Phi_e\}_{e \in \omega}$ via the following requirements.

\mathcal{R}_e : If $\text{range}(\Phi_e) = \text{range}(F)$, where $\Phi_e(x) := \lim_s \varphi_e(n, s)$, then $B = \Gamma^A$.

The strategy to meet a single requirement in isolation is as one would expect.

- (1) Choose a large integer z and put $f(e, s_0) = z$ with use $B \upharpoonright s_0$, where s_0 is the current stage. Keep $F(y) \neq z$ for $y > e$.
- (2) Wait for a column n with $\varphi_e(n, t_0) = z$.
- (3) Put $f(e, s_1) = z + 1$ with use $B \upharpoonright s_1$, where s_1 is the current stage.
- (4) Wait for $\varphi_e(n, t_1) = z + 1$ for some t_1 .
- (5) Release the $F(y) \neq z$ restraint and keep $F(y) \neq z + 1$ for $y > e$.
- (6) Wait for a $B \upharpoonright s_0$ change. While doing so, repeat Step 3 through Step 6 with $z + k$ replacing $z + 1$ and $z + k - 1$ replacing z for incrementally larger values of k .
- (7) If $B \upharpoonright s_k$ changes for some k , redefine $f(e, s) = z + k - 1$ for all $s \geq s_k$. Keep $f(y, s) \neq z + k$ for $y > e$.
- (8) If $\varphi_e(n, t) = f(y, s)$ for some $y > e$, return to Step 3 with y in place of e .

The key point is since $\mathbf{b} \neq \mathbf{0}$, then by standard Δ_2^0 permitting, eventually B will change below s_k for some k after the stage this use is declared. Thus \mathcal{R}_e is satisfied.

As with Theorem 2.2, the strategies combine without any difficulty on a tree. Lower priority strategies guess whether the outcome of higher priority strategies is finitary (i.e., $\Phi_e(n)$ is finite) or infinitary (i.e., $\Phi_e(n)$ is infinite). \square

7. OPEN QUESTIONS

We close with several questions (asked in numerous other places) that remain open.

Question 7.1. For which sets S is the strong η -representation of S computable?

Question 7.2. Which reduced abelian p -groups are computable? In particular, is there a reduced abelian p -group \mathcal{G} of length ω^2 for which there is no $\mathbf{0}^{(2^n)}$ -computable approximation function $f(i, x, s)$ with $u_{\omega \cdot i + F(i, x)}(\mathcal{G}) > 0$?

Question 7.3. What more can be said about possible limitwise monotonic spectra?

REFERENCES

- [1] Klaus Ambos-Spies, S. Barry Cooper, and Steffen Lempp. Initial segments of recursive linear orders. *Order*, 14(2):101–105, 1997/98.
- [2] C. J. Ash and J. Knight. *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 2000.
- [3] William W. Boone. Certain simple, unsolvable problems of group theory. I. *Nederl. Akad. Wetensch. Proc. Ser. A.*, 57:231–237 = *Indag. Math.* 16, 231–237 (1954), 1954.
- [4] William W. Boone. Certain simple, unsolvable problems of group theory. II. *Nederl. Akad. Wetensch. Proc. Ser. A.*, 57:492–497 = *Indag. Math.* 16, 492–497 (1954), 1954.
- [5] William W. Boone. Certain simple, unsolvable problems of group theory. III. *Nederl. Akad. Wetensch. Proc. Ser. A.*, 58:252–256 = *Indag. Math.* 17, 252–256 (1955), 1955.
- [6] William W. Boone. Certain simple, unsolvable problems of group theory. IV. *Nederl. Akad. Wetensch. Proc. Ser. A.* 58 = *Indag. Math.*, 17:571–577, 1955.
- [7] William W. Boone. Certain simple, unsolvable problems of group theory. V, VI. *Nederl. Akad. Wetensch. Proc. Ser. A.* 60 = *Indag. Math.*, 19:22–27, 227–232, 1957.
- [8] Wesley Calvert, Douglas Cenzer, Valentina Harizanov, and Andrei Morozov. Effective categoricity of equivalence structures. *Ann. Pure Appl. Logic*, 141(1-2):61–78, 2006.
- [9] Richard J. Coles, Rod Downey, and Bakhadyr Khoussainov. On initial segments of computable linear orders. *Order*, 14(2):107–124, 1997/98.
- [10] R. G. Downey. Computability theory and linear orderings. In *Handbook of recursive mathematics, Vol. 2*, volume 139 of *Studies in Logic and the Foundations of Mathematics*, pages 823–976. North-Holland, Amsterdam, 1998.
- [11] Rod Downey, Bakhadyr Khoussainov, Joseph S. Miller, and Liang Yu. Degree spectra of unary relations on (ω, \leq) . In *Logic, Methodology and Philosophy of Science*, Proceedings of the Thirteenth International Congress, pages 36–65. King’s College Publications.
- [12] L. Feiner. The strong homogeneity conjecture. *J. Symbolic Logic*, 35:375–377, 1970.
- [13] Andrey N. Frolov and Maxim V. Zubkov. Increasing η -representable degrees. Submitted.
- [14] Kenneth Harris. η -representation of sets and degrees. *J. Symbolic Logic*, 73(4):1097–1121, 2008.
- [15] Denis Hirschfeldt, Russell Miller, and Sergei Podzorov. Order-computable sets. *Notre Dame J. Formal Logic*, 48(3):317–347 (electronic), 2007.
- [16] Denis R. Hirschfeldt. Prime models of theories of computable linear orderings. *Proc. Amer. Math. Soc.*, 129(10):3079–3083 (electronic), 2001.
- [17] Asher M. Kach. Computable shuffle sums of ordinals. *Archive for Mathematical Logic*, 47(3):211–219, 2008.
- [18] Asher M. Kach and Joseph S. Miller. Embeddings of computable linear orders. In preparation.
- [19] Asher M. Kach and Daniel Turetsky. Limitwise monotonic functions, sets, and degrees on computable domains. *J. Symb. Logic*, to appear.
- [20] N. G. Khisamiev. The arithmetic hierarchy of abelian groups. *Sibirsk. Mat. Zh.*, 29(6):144–159, 1988.
- [21] N. G. Khisamiev. Constructive abelian groups. In *Handbook of recursive mathematics, Vol. 2*, volume 139 of *Stud. Logic Found. Math.*, pages 1177–1231. North-Holland, Amsterdam, 1998.
- [22] Bakhadyr Khoussainov, Andre Nies, and Richard A. Shore. Computable models of theories with few models. *Notre Dame J. Formal Logic*, 38(2):165–178, 1997.

- [23] Carolyn Knoll. *Degree Spectra of Unary Relations on (ω, \leq) and (ζ, \leq)* . M.S. in Mathematics, University of Waterloo, <http://hdl.handle.net/10012/4544>, 2009.
- [24] Manuel Lerman. On recursive linear orderings. In *Logic Year 1979–80 (Proc. Seminars and Conf. Math. Logic, Univ. Connecticut, Storrs, Conn., 1979/80)*, volume 859 of *Lecture Notes in Math.*, pages 132–142. Springer, Berlin, 1981.
- [25] P. S. Novikov. *Ob algoritmičeskoj nerazrešimosti problemy toždestva slov v teorii grupp*. Trudy Mat. Inst. im. Steklov. no. 44. Izdat. Akad. Nauk SSSR, Moscow, 1955.
- [26] Emil L. Post. Recursively enumerable sets of positive integers and their decision problems. *Bull. Amer. Math. Soc.*, 50:284–316, 1944.
- [27] Emil L. Post. Recursive unsolvability of a problem of Thue. *J. Symbolic Logic*, 12:1–11, 1947.
- [28] Matthew James S. Raw. *Complexity of Automorphisms of Recursive Linear Orders*. PhD in Mathematics, University of Wisconsin-Madison, 1995.
- [29] Joseph G. Rosenstein. *Linear orderings*, volume 98 of *Pure and Applied Mathematics*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982.
- [30] Maxim Zubkov. On η -representable sets. In *Computation and logic in the real world*, pages 364–366. 2007.

DEPARTMENT OF MATHEMATICS, STATISTICS, AND OPERATIONS RESEARCH, VICTORIA UNIVERSITY OF WELLINGTON, P.O. BOX 600, WELLINGTON 6140 NEW ZEALAND

E-mail address: Rod.Downey@ecs.vuw.ac.nz

DEPARTMENT OF MATHEMATICS, STATISTICS, AND OPERATIONS RESEARCH, VICTORIA UNIVERSITY OF WELLINGTON, P.O. BOX 600, WELLINGTON 6140 NEW ZEALAND

E-mail address: Asher.Kach@msor.vuw.ac.nz

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF WISCONSIN, 480 LINCOLN DRIVE, MADISON, WI 53706, USA

E-mail address: turetsky@math.wisc.edu