ITERATED EFFECTIVE EMBEDDINGS OF ABELIAN p-GROUPS

RODNEY DOWNEY, ALEXANDER G. MELNIKOV, AND KENG MENG NG

Abstract. Khisamiev [26, 31] and, independently, Ash, Knight, and Oates [2] characterized computable reduced abelian p-groups of finite Ulm type. The case of Ulm type ω has been an open problem for at least 20 years. The difficulty in the case of Ulm type ω is rooted in the uniformity of effective invariants (limitwise monotonic functions) corresponding to the computable group. Using iterated embeddings of p-basic trees, we construct a computable reduced abelian p-group of Ulm type ω having its invariants at the maximal possible level of complexity, the latter can be obtained from counting alternations of quantifiers.

The result gives an explanation of why the case of Ulm type ω seems unapproachable in general, while the techniques used in the proof may help to characterize computable members of some subclasses of type ω groups.

We also use the machinery of iterated embeddings of p-basic trees to solve a problem left open by Calvert, Cenzer, Harizanov and Morozov [5].

1. Introduction

Following Mal′cev [35] and Rabin [40], we say that a countable group H is *recursive* or *computable* if elements of H can be associated with natural numbers so that the group operation becomes a recursive function on these numbers. The above mentioned numbering of the group is called a *computable presentation* or *constructivistion* of the group. Equivalently, a group has a computable presentation if, and only if, the group admits an effective listing of its generators under which the word problem is solvable.

Mal′cev [35] initiated the systematic study of computable abelian groups. Among other results, Mal′cev characterized computable subgroups of $(\mathbb{Q}, +)$, and also showed that the additive group $\bigoplus_{i\in\omega} \mathbb{Q}$ admits more than one computable presentation, up to a computable isomorphism. After Mal′cev's fundamental paper [35], computable abelian group theory has been developing rapidly and simultaneously with other branches of effective algebra. These related branches include effective field theory (see Frölich and Shepherdson [16], Rabin [40], Metakides and Nerode [37]), computable Boolean algebras (Goncharov [19], Remmel [41]) and computable linear orders (Downey [12]). Other closely related subjects are the study of effectively presented vector spaces [8, 9, 36] and the theory of computable ordered groups [13, 20]. For early developments in the field of computable abelian groups, see Nurtazin [38], Smith [43], Lin [34], and Khisamiev [25].

Modern theory of computable abelian groups uses methods of abelian group theory [17, 18, 29], pure computability theory [44] and computable model theory [3, 15], as well as tools specific to the field [32, 11, 1]. Standard references for the theory of computable ablelian groups are [32, 14]. Reflecting the situation in classical algebra, methods used in the theory of computable abelian p-groups [32] tend to be different from techniques applied in the torsion-free case [11, 1, 30]. In contrast to torsion-free groups, algebraic properties of countable abelian p-groups are very well understood. Thus, the study of computable abelian p-groups would unlikely require significantly new algebraic methods. However, even a masterful use of known algebraic invariants may not be sufficient, since most of the difficulties that occur are computability-theoretic in nature. As we will see, even characterizing computably presented direct sums of cyclic groups already requires some new computability-theoretic ideas. For a systematic development of computable abelain p-group theory, see a survey paper of Khisamiev [32]. See also [4, 39, 5] for some further results in this direction.

In this paper, we look at the effective content of the classical result of Ulm (see [29]). In the subsection below, we briefly discuss the well-known algebraic concepts which are central to the paper. A reader familiar with Ulm's classification of countable reduced abelian p-groups may skip the subsection below.

1.1. **Reduced abelian p-groups.** Throughout the paper, all groups are countable and abelian. Let p be a prime number. A non-zero element g of A has infinite height if for every k the equation $p^k x = g$ has a solution in A. Elements of infinite height generate a sub-group A' of A. Iterating this process, we can define $A^{(\alpha)}$ for every ordinal α. The quotients $A_0 = A/A'$ and $A_\alpha = A_\alpha/A^{(\alpha+1)}$ contain only (non-zero) elements of finite height. Since A is countable, there must be a countable α for which

$$A^{(\alpha)} = A^{(\alpha+1)}.$$

The least such α is called the Ulm type of A and is denoted by $u(A)$ in this paper. If $A^{(u(A))} = \mathbf{0}$, then A is *reduced*.

One can show that the *Ulm factors* A_α are simply direct sums of finite cyclic groups. Such a direct sum can be fully classified by the sizes of its elementary cyclic summands. This classification can be generalized:

Theorem 1.1 (Ulm). *The isomorphism type of a countable reduced (abelian) p-group is completely determined by the isomorphism types of its Ulm factors.*

In this paper, we addresses the long-standing question of which Ulm invariants correspond to computable reduced abelian p-groups.

1.2. **Computable groups of finite Ulm type.** Khisamiev [26] was the first to discover that computable reduced abelian p-groups of Ulm type 1 (those being direct sums of cyclic groups) can be completely characterized using the concept of a *limitwise monotonic function.*

Definition 1.2 (Khisamiev [26]). A total function F is *limitwise monotonic* if $F = \lambda x. \sup_y f(x, y)$, where f is computable.

If we replace f by a $0^{(n)}$-computable function in the definition above, we obtain the notion of $0^{(n)}$-limitwise monotonicity. A set is $0^{(n)}$-*limitwise monotonic* if it is the range of a $0^{(n)}$-limitwise monotonic function. For infinite Σ^0_{n+1} sets, the latter is equivalent to *containing* an infinite range of a $0^{(n)}$-limitwise monotonic function (see, e.g., [28, 21]). Khisamiev [26] proved:

Theorem 1.3. *Let A be a direct sum of finite cyclic p-groups whose orders are unbounded. Then A has a computable copy if, and only if, the following two conditions hold:*

(1) *$S(A) = \{(m, k) :$ at least k summands of A have order $p^m\}$ is a Σ^0_2-set, and*
(2) *$\#A = \{m : Z_{p^m}$ is a summand of $A\}$ is limitwise monotonic.*

The concept of limitwise monotonicity was new to computability theory. Limitwise monotonic functions have found various applications outside the theory of computable groups [7, 21, 27, 22, 6, 23, 24, 10]. We also mention that Khoussainov, Nies, and Shore [33] independently introduced limitwise monotonic functions in the context of computable model theory.

Theorem 1.3 can be generalized to any finite Ulm type.

Theorem 1.4. [31, 2] *Let A be a reduced (abelian) p-group of Ulm type $n < \omega$. Then the following are equivalent:*

(1) *A has a computable copy;*
(2) *A can be represented by a computable p-basic tree;*
(3) (a) *for every $i < n$, the set*
$$S(A_i) = \{(m, k) : \text{ at least } k \text{ summands of } A_i \text{ are of order } p^m\}$$
is Σ^0_{2i+2}, and
(b) *for every $i < n$, the set*
$$\#A_i = \{m : Z_{p^m} \text{ is a summand of } A_i\}$$
is $0^{(2i)}$-limitwise monotonic.

1.3. **The case of Ulm type ω.** It is not known if Theorem 1.4 can be generalized to groups of Ulm type $\geq \omega$. One direction of the proof of Theorem 1.4 uses a non-uniform argument. In fact, the non-uniformity seems to be the crucial obstacle when one attempts to generalize Theorem 1.4 to groups of type ω. It is well-known though that the theorem holds for A of Ulm type ω if the limitwise monotonic functions ranging over $\#A_i$ are given uniformly.

One can observe that the sets $\#G_i$ have to be limitwise monotonic, uniformly in i or not, if G is computable. Counting the number of quantifiers (see Fact 3.1) shows that finding an index for a limitwise monotonic function ranging over $\#G_i$ is at most Π^0_3 over $0^{(2i)}$. So, for instance, $\#G_0$ requires at most a Π^0_3-guessing procedure. In the theorem below we show that this upper bound (namely, $\Pi^0_{(3+2i)}$) is sharp.

Theorem 1.5. *The exists a computable reduced abelian p-group G of Ulm type ω such that there is no uniformly $\Sigma^0_{(3+2i)}$-effective procedure guessing indices for limiwise monotonic functions ranging over $\#G_i$.*

The following immediate corollary proves a conjecture of Ash, Knight and Oates [2]:

Corollary 1.6. *There exists a computable reduced abelian p-group G of Ulm type ω for which $\#G_i$ are not uniformly $0^{(2i)}$-limitwise monotonic.*

Theorem 1.5 also gives an evidence that the question of which computable reduced p-groups are computably presentable is as difficult as it could be. We have already observed that it is *necessary* that the sets $\#G_i$ are limitwise monotonic (relative to $0^{(2i)}$). Any characterization would have to use this property to pass from the collection of the invariants to a computable group, unless it turns out that very few Ulm invariants may correspond to computable groups. The general expectation is that the latter is not the case, and whence any proof of the positive direction would have to incorporate an iterated $0'''$-construction with various combinatorial and algebraic difficulties (to be discussed in Conclusion).

The key technical tool of the present paper is contained in Proposition 3.5 which gives a uniform embedding performed on top of a strategy potentially having a Π^0_3-outcome. Some elements of the $0'''$-machinery as well as specific purely algebraic techniques are vital for both construction and its verification.

1.4. **A categoricity question.** Recall that a computable structure is Δ^0_α-categorical if any two computable isomorphic copies of the structure are Δ^0_α-isomorphic [3]. Calvert, Cenzer, Harizanov, and Morozov asked (see Problem 5.1 in [5]):

Question 1.7. *Let G be a computable abelian p-group isomorphic to $D \oplus H$, where D is a direct sum of finitely many copies of the Prufer group Z_{p^∞}, and H is a direct sum of cyclic summands of unbounded orders. Can G be Δ^0_2-categorical?*

We answer the question in negative. In fact, we prove more:

Theorem 1.8. *Let G be a computable p-group of finite Ulm type n, such that:*
(1.) $G^{(n)} \cong \oplus_{j \leq m} Z_{p^\infty}$, for some $m \in \omega$;
(2.) orders of cyclic summands in G_{n-1} are not bounded.
Then G is not Δ^0_{2n}-categorical.

In the special case when $n = 1$ we get exactly groups satisfying conditions of Question 1.7. The theorem improves earlier results of Dushenin [] who used complex full approximation techniques to construct non-Δ^0_{2n}-categorical groups in these classes for small n.

2. Background and conventions

We assume that the reader has a sufficient background in computability theory [44] and computable model theory [3, 15]. We will be using some rudiments of abelian group theory, standard textbooks are [17, 18, 29], but no background in abelian group theory is assumed.

2.1. ***p*-Basic trees.** In mathematical practice, it is convenient to use tree-like diagrams representing abelian p-groups. For instance, imagine a tree on vertices v_1, v_2, v_3 and v_4 such that v_1 is the root having successor v_2, and v_3, v_4 are the only two children of v_2. This tree corresponds to the abelian group

$$B = \langle v_1, v_2, v_3, v_4 : v_1 = 0, pv_2 = v_1, pv_3 = v_2, pv_4 = v_2 \rangle.$$

Notice that the same group can be represented by another tree: for instance, pick v_1, v_2, v_3 and $v_3 - v_4$ as new generators. Both trees are called *p-basic trees* of B [42]. The formal definition is:

Definition 2.1. [42] A p-basic tree is a set X together with an binary operation $\cdot$ of the sort $\{p^n : n \in \omega \setminus \{0\}\} \times X \to X$ such that:
(1) there is a unique element 0 in X for which $p \cdot 0 = 0$, and
(2) for each nonzero element x in X, there is a positive integer n such that $p^n \cdot x = 0$.

Given a p-basic tree X we can pass to an abelian p-group $G(X)$. We make the set $X \setminus \{0\}$ the set of generators, and put $px = y$ into the collection of relations if $p \cdot x = y$ in X. Every countable reduced abelian p-group is generated by some well-founded p-basic tree [42].

Convention 2.2. We typicaly identify a p-basic tree with the corresponding abelian p-group, but the reader should keep in mind that the choice of a p-basic tree is typically not unique.

Recall the notion of ordinal tree rank for a well-founded tree: every leaf has tree rank 0, and the tree rank of any other vertex is the least ordinal greater than the ranks of all its successors. Notice that every element in $G(X)$ can be uniquely represented in the form $\sum_i m_i v_i$, where $v_i \in X$ and $m_i \in \{1, \dots, p-1\}$.

Definition 2.3. Suppose X is a well-founded p-basic tree, and $G(X)$ is the corresponding group. The rank of $\sum_i m_i v_i$, where $v_i \in X$ and $m_i \in \{1, \dots, p-1\}$, is the minimum of tree ranks of the v_i in X.

The definition is independent on the choice of the underlying p-basic tree (follows from Proposition 1 of [42]). We will use the following consequence of Definition 2.3 without explicit references:

Remark 2.4. The collection of tree-ranks that appear in X is the same as the collection of ranks that appear in $G(X)$.

A non-zero element has rank $k \in \omega$ if, and only if, it has height k. Non-zero elements having rank $\geq \omega$ are exactly the elements of infinite height. Thus, we could define the Ulm factors using ranks rather than heights. Furthermore, the Ulm invariants of $G(X)$ can be reconstructed using only tree ranks which appear in X [42].

2.2. **Trees which give rise to isomorphic groups.** We will not completely describe the congruence relation $\sim$ on trees defined by the rule $T \sim X$ iff $G(T) \cong G(X)$. A detailed analysis of $\sim$ can be found in [42]. We will be using some weaker sufficient conditions and partial invariants.

In a tree, a chain is called *simple* if each node in the chain has at most one successor. Consider the following example.

Example 2.5. Suppose we have a tree T, and suppose $v \in T$ has tree rank α. Suppose also w is a successor of v having rank $k \in \omega$ such that $k + 1 < \alpha$. Suppose further that w is on top of a simple chain $c(w)$ (there are no splittings below w). Remove the chain $c(w)$ together with w from T, and then adjoin $c(w)$ to 0. The resulting p-basic tree X has the property $G(T) \cong G(X)$.

The procedure described in Example 2.5 is called *stripping*. We can iterate this process and obtain a *fully stripped* tree representing the same group. The only restriction is that we have to keep *some* sequence below a node witnessing its tree-rank. For example, a fully stripped tree for a group of Ulm type 1 is simply a collection of simple chains attached to 0. We have just proved that every countable reduced p-group of Ulm type 1 is isomorphic to a direct sum of cyclic p-groups [17].

Another very special case of the general framework on p-basic trees is stated in the fact below.

Fact 2.6. *Suppose T and X are p-basic trees so that 0 has rank ω in X. There exists a p-basic tree V such that $G(V)_0 \cong G(X)$ and $G(V)' \cong G(T)$.*

Proof Sketch. Attach infinitely many finite simple chains to every node in T making ranks of vertices in (the image of) T infinite within V. The lengths of the finite chains should be based on the ranks which occur in X. □

Although classically Fact 2.6 is a triviality, an effective analog of it is not straightforward and is the main technical tool of [2, 31]:

Lemma 2.7. *Let T be a computable p-basic tree of Ulm type 1 in which 0 has tree-rank ω, and let C be any Π^0_2 subtree of $\omega^{<\omega}$ (C is viewed as a p-basic tree). There exists a computable p-basic tree U expanding C such that $U_0 \cong T$ and $U' = C$.*

Proof Sketch. First, observe that there exists a computable limitwise monotonic function f such that $\sup_y f(x, y)$ is infinite and ranges over lengths that occur in T. Using this limitwise monotonic function, as well as the fact that the Ulm invariant of T is naturally represented as a Σ^0_2 (multi)set, we shall imitate the proof of Fact 2.6. If a Π^0_2-predicate fires on $\sigma \in \omega^{<\omega}$, as well as on all initial segments of σ, we add more simple chains below σ using f. Without loss of generality, we may assume that $\lambda x \sup_y f(x, y)$ is injective and even that $\sup_y f(x, y) \geq n_x$ for any computable increasing sequence $(n_x)_{x\in\omega}$.

There are several difficulties in the construction that need to be addressed:

(1) *Utilizing finite subtrees.* If the Π^0_2 predicate representing C does not fire anymore on σ, we need to ensure the finite tree that we have built on top of σ (for the sake of approximating C) does not give finite chains of wrong lengths after stripping. (There could have been σ' extending σ which we thought was in C, say.) Since C contains $\emptyset$, there exists a $\tau \subseteq \sigma$ such that $\tau \in C$. Thus, we may assume that, whenever a finite tree is abandoned due to the predicate being not active, the longest chain present in this subtree can be stripped off (see Example 2.5). The difficulty is that the longest chain present in the abandoned finite subtree may not be in the range of the limitwise monotonic function. We then pick x so that $\sup_{t\leq s} f(x, t)$ is large enough and *extend* that longest chain using this function. We may organize the construction so that, even though $\sup_{t\leq s} f(x, t)$ is large, the corresponding string $\sup_t f(x, t)$ can still be stripped off at the end. We do so by using very long strings when approximating initial segments of σ.

(2) *Resurrecting σ.* If C becomes active on σ again, we need to start adding longer and longer finite chains below σ making progress in showing that $\sigma \in U'$. Nonetheless, due to σ being inactive at previous stages, we have extended the longest chain below σ to a larger value $\sup_{t\leq s} f(x, t)$. If we keep it like that, and simply add an even longer simple chain below σ, we might be in trouble since this size is not present in T. We could extend that previously extended longest string once again, using a new argument of the limitwise monotonic function. We then argue that the construction can be organized so that each string can be further extended at most twice.

(3) *Multiple sizes in T.* If T has exactly 3 simple chains of size 4 (corresponding to 3 elementary summands of type Z_{p^4}), we have to use exactly 3 simple chains of that length when constructing U. This is not really a problem, since we have already observed the Ulm invariant of T is Σ^0_2. All we need to do is just tracing which sizes we have used to approximate C, and the rest we could realize as simple chains adjoined to $\emptyset$. If m leaves the Ulm

invariant (Σ^0_2-process), we use f on an appropriate argument to grow the simple chain of length m, attached to $\emptyset$, to a safe length.

The proof does not contain any further significant difficulties. $\square$

3. Proof of Theorem 1.5

Throughout the proof, all groups are reduced abelian p-groups. Also, we will typically identify a p-basic tree and the group it generates, but the reader should keep in mind that non-isomorphic trees may generate isomorphic groups.

Preliminary analysis. Recall that for a group G, the set of finite heights which occur in G_i is denoted by $\#G_i$.

Fact 3.1. *In a computable G of Ulm type ω the multisets $\#G_i$ are uniformly $\Pi^0_3(0^{(2i)})$-limitwise monotonic. (There exists a $\Pi^0_{(3+2i)}$-sequence of sets of indices witnessing limitwise monotonicity.)*

Proof of Fact 3.1. To say that an element $g \neq 0$ has infinite height in G takes 2 quantifiers. We also say that there is no $h \in G$ having infinite height such that $ph = g$. The combined complexity is Π^0_3. Given such an element, we can effectively pass to a limitwise monotonic function with the range $\{n : (\exists a)(h_p(a) = 0 \wedge p^n g = a)\} \subseteq \#G_0$. It is well-known that one can effectively pass from a limitwise monotonic function on an infinite subset to a limitwise monotonic function ranging over the whole set. A relativized version of this argument gives $\Pi^0_3(0^{(2i)})$ when considering $\#G_i$. $\square$

We aim to construct a computable reduced abelian p-group G of Ulm type ω for which the indices of functions witnessing limitwise monotonicity of $\#G_i$ are not uniformly $\Sigma^0_3(0^{(2i)})$.

Notations and requirements. Although it is intuitively clear which requirements we need to satisfy, we prefer to formally state them. Let $(F_{0,j})_{j\in\omega}, (F_{1,j})_{j\in\omega}, \ldots$ be the effective listing of all uniformly c.e. sequences of predicates. Based on this listing, and using alternating projections and complementations, we can associate every (uniformly) $\Sigma^0_{(3+2i)}$ sequence of predicates with a single index e, and denote it $(R_{e,i})_{i\in\omega}$, where $R_{e,i}$ is Σ^0_{3+2i} uniformly in i. We say that j witnesses $0^{(e)}$-limitwise monotonicity of a set X if $f(x) \leftrightharpoons \sup_y \Phi_j(0^{(e)}; x, y)$ is total, $\mathrm{rng}_x f(x)$ is infinite, and

$$\mathrm{rng}_x f(x) \subseteq X.$$

A set S witnesses $0^{(e)}$-limitwise monotonicity of X if each $j \in S$ witnesses $0^{(e)}$-limitwise monotonicity of X. We meet, for every e, the requirement:

$$L_e : R_{e,3e} \text{ does not witness } 0^{(6e)}\text{-limitwise monotonicity of } \#G_{3e}.$$

The reason we are using G_{3e} instead of G_e is related to the outcomes of the basic diagonalization strategy and will be explained later. In subsection 3.1 we describe one diagonalization strategy in isolation. Then, in subsection 3.2, we modify the basic strategy to a strategy which can handle any finite tree. In subsection 3.3 we merge the modified diagonalization strategy and a Σ^0_3-guessing procedure. In subsection 3.4 we construct an auxiliary group F that will be used later in the construction of G. In subsection 3.5, we describe how to build G assuming certain uniform operators exist. The operators are constructed in subsection 3.6. The proof is finished in subsection 3.7.

3.1. **The basic strategy.** Recall that for a group G, the collection of finite heights which occur in G_0 is denoted by $\#G_0$. Suppose we wish to uniformly construct a computable group G in which $\#G_0$ is infinite *and* not limitwise monotonic via $\sup_y f(x,y)$ for a given (partially) computable f:

$$\#G_0 \neq range\, \lambda x. \sup_y f(x,y).$$

We initially start with the computable group $\bigoplus_{m,n\in\omega} Z_{p^n} a_{m,n}$, where $\{a_{m,n} : m,n \in \omega\}$ is a computable collection of distinguished generators of its cyclic summands. We can think of this group as of a fully stripped tree with 0 having rank ω such that every size is represented by infinitely many simple chains attached to the root.

Note 3.2. We additionally assume that we know in advance the length of any simple chain in the tree representing $\bigoplus_{m,n\in\omega} Z_{p^n} a_{m,n}$ (equivalently, we know orders of the elementary summands in advance).

We wait for s and x such that $\sup_{y\leq s} f_s(x,y)$ is defined and is equal to $k > 1$. If we never see such a computation, then either f is not total or the range of $\lambda x. \sup_y f(x,y)$ is a finite set, and we are done. If we see such s and x, we add extra elements to the group as follows:

(1) for every elementary cyclic summand $Z_{p^k} a_{m,k}$ of order k, pick its generator $a_{m,k}$;
(2) put the label $\boxed{\text{f}}$ onto $pa_{m,k}$;
(3) introduce a new element $h_{m,k}$ and declare $p^2 h_{m,k} = pa_{m,k}$.

Notice that $h_{m,k}$ has order p^{k+1} in the group. Also, the subgroup generated by $\{a_{m,n}, h_{m,k}\}$ can be represented as

$$Z_p(a_{m,k} - ph_{m,k}) \oplus Z_{p^{k+1}} h_{m,k}.$$

Thus, the cyclic summand Z_{p^k} is currently not present in the complete decomposition of G_0. Therefore, if $\sup_y f(x,y) = \sup_{y\leq s} f_s(x,y)$, then R_e is met. If at a later stage $t > s$ we see $\sup_{y\leq s} f_s(x,y) < \sup_{y\leq t} f_t(x,y)$, we need to consider the following cases:

Case 1. $\sup_{y\leq t} f_t(x,y) = k+1$. For every element carrying $\boxed{\text{f}}$ (these currently are $pa_{m,k}$), introduce a new generator $h'_{m,k}$ and declare $p^2 h'_{m,k}$ equal to the element carrying $\boxed{\text{f}}$.

Case 2. $\sup_{y\leq t} f_t(x,y) = k' > k+1$ (recall $k = \sup_{y\leq s} f_s(x,y)$). In this case we remove all labels $\boxed{\text{f}}$ which were put by the strategy at stage s. Repeat the $(1)-(3)$ with k replaced by k'. (Notice that in this case we put new labels onto $pa_{m,k'}$.)

If the strategy introduces new elements to the group at stage v, we say that the strategy *is active* at stage v. If the strategy has been active, it has a witness x an argument of the potential limitwise monotonic function. The general strategy proceeds iterating the actions explained above.
The strategy has the following outcomes:

$\mathbf{s}_n$: The strategy is temporarily not active, and it has been active at n-many stages.
∞: The strategy is temporarily active.

If the strategy eventually settles at $\mathbf{s}_n$ for some n, then $G' = \mathbf{0}$ and

$$\#G_0 = \omega \setminus \{\sup_y f(x,y)\}.$$

If the strategy is active infinitely often, then we will build a computable group G such that

$$\#G_0 = \omega.$$

It is important that in this case we may have $G' \neq \mathbf{0}$, but in any case $G'' = \mathbf{0}$. Also, in any case we have $G_0 = G_0 \oplus G_0$; in other words, every $n \in \#G_0$ is represented by infinitely many elements.

The isomorphism type of G depends on f and can not be effectively predicted in general. Notice that the strategy enumerates, with all possible uniformity, a group having a computable p-basic tree. *We may assume that the strategy actually builds a tree rather than the corresponding group.*

3.2. **The modified strategy.** We are aiming to build a group having a computable p-basic tree. The basic strategy described in the previous subsection will not be sufficient for this goal, since other strategies may possibly effect the isomorphism type of G_0 producing "junk". More specifically, we need to explain the more complicated case when the basic strategy needs to deal with an arbitrary finite tree, not simply with a finite simple chain. This situation will occur in Proposition 3.5 which is the key technical tool of the paper.

We initially start having the fully stripped p-basic tree for $\bigoplus_{m,n\in\omega} Z_{p^n} a_{m,n}$ as an input of the procedure. At a later stage, some other strategy may add a finite p-basic tree V (or several finite trees) which will contribute to the isomorphism type of G_0. We need to satisfy

$$\#G_0 \neq range\, \lambda x. \sup_y f(x,y).$$

For the sake of this goal, the strategy may start adding finite chains to some elements of V. As a consequence of its actions on V, the construction will produce a potentially infinite tree T on top of the V-component. At stage s, we will have a finite tree $T[s]$, and $T = \bigcup_s T[s]$. The key idea is:

Never add extra chains to elements from $T[s] \setminus V$.

If we can implement this idea, only elements of V will possibly have infinite heights in G. Since V is finite, we will have $G'' = \mathbf{0}$.

Recall that every finite tree can be transformed to a collection of finite chains growing from a single root 0. It is possible to effectively trace images of finite chains under such a transformation and see which chains contribute to the collection of heights realized in $T[s]$. We say that $g \in T[s]$ is *dangerous* if

(1) g is a terminal node, and
(2) there exists a finite chain which terminates at g and witnesses that

$$l_{f,s} = \sup_{y \leq s} f_s(x,y)$$

is realized as a height of some element in (the group generated by) $T[s]$.

Modified strategy restricted to $T[s]$. If $g \in T[s]$ is dangerous, then consider the cases:

Case 1. We have $g \in V$. Then add a new element x to $T[s]$ and declare $px = g$.

Case 2. We have $g \in T[s] \setminus V$, and there exists m such that $h = p^{m+1}g$ is in V but $p^m g \in T[s] \setminus V$. Add a new element x to $T[s]$ and declare $p^{m+2}x = h$ (thus also adding $p^k x$ for $0 < k \leq m+1$).

In both cases, declare g not dangerous. Once there are no dangerous elements left, go to the next stage.

Verification. Note that g can not represent height $l_{f,s}$ in T_0. The action adds a chain which either extends g by 1 point (Case 1) or represents the new relation $h = p^{m+1}g$ (Case 2). In the first case g is not an end-point anymore, and can not represent any finite height in $\#T_0$ itself anymore. In the second case, notice that $h = p^{m+1}g$ is a terminal node in V. Thus, in the second case g belongs to a simple chain of length less than $l_{f,s}$, and will represent a direct summand of order smaller than $l_{f,s}$ (see the preliminary section or the example from the previous sub-section).

We could argue (possibly modifying the strategy) that the element x can not become dangerous unless l_f changes. But notice that even if x was dangerous without l_f changing, we would repeat the strategy above with x in place of g, using the same h. Eventually we would add a chain of length $l_{f,s}$ below h. Since h is a terminal node in V, that new added chain can no longer represent height $l_{f,s}$. Consequently, its end-vertex can not be declared dangerous unless l_f increases.

We conclude that eventually no element can represent $l_{f,s}$, unless the latter increases to a new value. If l_f ever stabilizes, we end up with a finite tree T such that $\lim_s l_{f,s} \notin \#T_0$. If l_f keeps increasing forever, we end up constructing a (possibly infinite) tree T containing V such that only elements of the *finite* V can possibly have infinite height. Thus, heights of elements in T' are bounded in T', and consequently $T'' = \mathbf{0}$.

3.3. **The strategy combined with a Σ^0_3-guessing.** In this subsection we explain how we diagonalize against a single Σ^0_3 predicate. All procedures in the subsection are effective. In general, we will be relativizing to an appropriate oracle.

Given a Σ^0_3-predicate represented in the form $\{e : \exists x \exists^\infty y U(x, y, e)\}$, where U is c.e., we will guess which pair $\langle e, x\rangle$ is least such that $\exists^\infty y U(x, y, e)$ (if there is any). Each pair $\langle e, x\rangle$ will be associated with a basic diagonalization strategy working with the function having index e. At stage s the basic module associated with $\langle x, e\rangle$ will be working within interval $I_{e,x}[s]$ of size at least $\sup\{n : \exists^n y \leq s\, U_s(x, y, e)\}$ (i.e., the interval is increased if the predicate "fires" again). At stage s we have a partitioning of ω into sub-intervals:

$$I_{0,0}[s], I_{0,1}[s], I_{1.0}[s], \ldots,$$

from left to right, where

$$I_{a,b}[s] = [\, m_{a,b}[s], n_{a,b}[s]\,].$$

We may additionally assume that if $\langle c, d\rangle = \langle a, b\rangle + 1$ then $m_{c,d}[s] \geq n_{a,b}[s] + \langle a, b\rangle + 1$, so that the intervals are sufficiently far apart even if the predicate does not "fire" for some pairs.

The basic strategy associated with $\langle e, x\rangle$ will also be aiming to introduce its witness, a natural number $l_{e,x,s}$ representing the supremum of $\Phi_e(\cdot,\cdot)$ on some (first found) input (z, w) such that $\Phi_e(z, w) \geq m_{e,x}[s]$ and set

$$l_{e,x,s} = \inf\{\sup_{z \leq s} \Phi_e(z, w), n_{e,x}[s]\}.$$

We initialize the basic module associated with $\langle e, x\rangle$ if one of the modules with smaller index increases its interval. In this case $m_{e,x}$ will be lifted to a fresh large number (larger than any number seen so far in the construction) and $l_{e,x,s}$ will be set undefined.

We visualize the configuration at a stage s as follows. We have intervals corresponding to the influence of each sub-strategy, and *labels* $l_{e,x,s}$ representing lengths which need to be avoided when constructing a tree. Notice each label may move only to a larger value, and every time one of the interval increases in size all larger labels will be removed and then possibly put on numbers which are very large. (It is crucial for the construction.)

3.4. **The definition of F.** We describe the group F build by the procedure (combined with a Σ^0_3-guessing) on input $\bigoplus_{m,n\in\omega} Z_{p^n} a_{m,n}$ represented by a tree consisting of simple chains growing from the root. We additionally assume that the lengths of simple chains in the tree representing $\bigoplus_{m,n\in\omega} Z_{p^n} a_{m,n}$ are known in advance (Note 3.2).

Before stage s begins we have a finite collection of finite trees $\{V_i[s-1] : i \leq s-1\}$. The tree $V_i[s-1]$ contains either one of the $a_{m,n}$ or is built around a newly introduced simple chain. All $V_i[s-1]$ share the same root, in other fords, the corresponding group is a direct sum of groups corresponding to the $V_i[s-1]$.

Construction. At stage s, let each of the sub-strategies indexed by pairs $\langle e, x\rangle \leq s$ act on $V_i[s-1]$, for each $m, n \leq s$, according to the instructions given in subsection 3.2. One extra restriction is that the basic strategy associated with $\langle e, x\rangle$ is not allowed to use simple chains of sizes $l_{e',x',s}$, $\langle e', x'\rangle < \langle e, x\rangle$, all other sizes are available[1]. If the label $l_{e,x}$ is removed or is put onto a larger number, we introduce infinitely many simple chains representing this currently unoccupied length and attach it to the root of the tree.

The following outcomes are possible:

- (e, x, ∞): The interval $I_{e,x}$ grows to infinity with eventually stable left-most point, and the eventually stable witness corresponding to $\langle e, x\rangle$ tends to infinity.
- (e, x, k): The interval $I_{e,x}$ grows to infinity with eventually stable left-most point, and the eventually stable witness corresponding to $\langle e, x\rangle$ is stuck at k. It includes the case when eventually no witness can be chosen (an outcome of the form (e, x, f) with symbol f).
- g: This Π^0_3-outcome is a global win corresponding to all intervals being eventually finite.

If (e, x, ∞) is the true outcome, no labels to the left of $I_{e,x}$ ever move after a stage s. At every stage $t \geq s$ at which $I_{e,x}$ increases in size, all labels of sub strategies associated with larger pairs will be moved beyond $I_{e,x}$ to fresh large numbers. In fact, they will be lifted up so large that no tree among $V_{m,n}$ which ever was influenced by their actions will ever be modified by these strategies again. Consequently, we are in the situation similar to the one described in subsection 3.1, but we need to incorporate modifications contained in subsection 3.2 to see that in the limit we construct a p-basic tree F with $F'' = 0$.

[1] Recall that intervals and, whence, labels corresponding to different strategies are sufficiently far apart.

If (e, x, k) is the true outcome, we will end up with a p-basic tree F such that $\#F_0 = \omega \setminus S$, where S is a finite set containing all eventually stable l-labels. In fact, $F' = 0$ in this case. Similar argument applies when the true outcome is g, but in this case $\#F_0 = \omega \setminus S$ where F is potentially infinite. In this case we again have $F' = 0$. (Recall that the intervals are sufficiently far apart, thus we do not have the situation when one l-label is an immediate successor of another l-label, say.)

In any of these cases we succeed in constructing a p-basic tree avoiding an index from the given Σ^0_3-set.

3.5. **Describing G.** Recall that we need to construct a computable reduced abelian p-group G of Ulm type ω and meet:

$$L_e : R_{e,3e} \text{ does not witness } 0^{(6e)}\text{-limitwise monotonicity of } \#G_{3e},$$

where the $R_{e,i}$ are Σ^0_{3+2i} uniformly in i.

The strategy for L_e is similar to the one discussed in the previous subsection. We run this strategy effectively in $0^{(6e)}$. As a result, L_e will uniformly produce a $0^{(6e)}$ -computable p-basic tree $A(3e)$ having Ulm type either 1, 2 or 3, depending on the true outcome of L_e.

We wish to construct computable G such that $G_{3e} = A(3e)_0$, for every e. If we succeed, then R_e will be met for every e. We will also make sure $G_{3e+1} \cong G_{3e+2} \cong \bigoplus_{m,n} Z_{p^m} a_{m,n}$ for every e.

Why do we homogenize G_{3e+1} and G_{3e+2}? The reason is that, depending on the outcome of L_0, we may or may not have elements of infinite height in $A(0)$. Consequently, a plane construction without homogenizing would have to sort out conflicts between R_0 and R_1, if R_1 were to work within G_1. Furthermore, we will need to put all groups together in a tower (this is the main difficulty), and in some cases we may have "junk" which will result elements of high rank. We will show that the ranks of the "junk" elements will be less than $\omega \cdot 3$, and they could potentially effect at most two more strategies. We circumvent this potential difficulty homogenizing G_1 and G_2 and working within G_3 for the sake of L_1.

How do we construct G? Firstly, we are going to construct not G itself but a computable p-basic tree representing G. Throughout the rest of the paper, we may (classically) identify this p-basic tree with the corresponding p-group. Secondly, instead of constructing the whole G at once, we will construct a uniformly computable sequence of p-basic trees $(B(i))_{i\in\omega}$ such that $B(i)$ is of Ulm type i. We will have $B(i)_{3e} \cong A(3e)$ and $B(i)_{3e-1} = B(i)_{3e-2} \cong \bigoplus_{m,n\in\omega} Z_{p^m} a_{m,n}$, for every $3e \leq i$. We will set $G = \bigoplus_{i\in\omega} B(i)$.

How do we build $B(i)$? We construct $B(i)$ using operators which map Π^0_2-subtrees of $\omega^{<\omega}$ to computable trees. Recall that every Δ^0_2-tree is isomorphic to a Π^0_1-subtree of $\omega^{<\omega}$, with all possible uniformity.

Given a Π^0_2 p-basic tree D, we can uniformly produce a a computable p-basic tree H such that $H_0 \cong \bigoplus_{m,n} Z_{p^m} a_{m,n}$ and $G' \cong D$ (see Proposition 3.4). We will also prove that, given the computable tree F constructed by one of the diagonalization strategies (think of R_0 and $A(0)$) and a Π^0_2 subtree C of $\omega^{<\omega}$ such that $C_0 \cong C_1 \cong \bigoplus_{m,n} Z_{p^m} a_{m,n}$, we can uniformly construct a computable tree U such that $U_0 \cong F_0$ and $U' \cong C$ (see Proposition 3.5). Assuming these operators exist, we can uniformly construct the trees/groups $B(i)$, and then uniformly pass to $G = \bigoplus_{i\in\omega} B(i)$. The group G will have the desired properties.

We may summarize subsection 3.3 as follows:

Fact 3.3. *There exists a uniform procedure which, given a Σ^0_3 predicate R, produces a computable p-basic tree F of Ulm type at most 2 such that:*

a. $F_0 \cong F_0 \oplus F_0$ *(thus, every finite height is represented in F_0 by infinitely many elements);*
b. *R is not a collection of indices of functions witnessing limitwise monotonicity of $\#F_0$.*

3.6. **The operators mapping Π^0_2-trees to computable ones.** Recall that we identify a p-basic tree with the group it generates. We will need the following two propositions. The first proposition is a special case of a known technical result [2, 31] (also will be stated in Lemma 2.7 below), the second proposition is new.

Proposition 3.4. *Given a Π^0_2 p-basic tree D, we can uniformly produce a computable p-basic tree H such that $H_0 \cong \bigoplus_{m,n\in\omega} Z_{p^m} a_{m,n}$ and $H' \cong D$.*

Proof. Adjoin more finite chains of greater length below x is there is more evidence that $x \in D$. Since *all* finite lengths amy occur, no further work needs to be done. Also immediately adjoin infinitely many chains of each finite size to the root. □

Throughout this subsection, the p-basic tree from Fact 3.3 will be called *R-avoiding* and will be denoted F.

Proposition 3.5. *There exists a uniform procedure which, given a Σ^0_3 predicate R, the procedure building a computable R-avoiding F, and a Π^0_2 p-basic tree C such that $C_0 \cong C_1 \cong \bigoplus_{m,n\in\omega} Z_{p^m} a_{m,n}$, produces a computable p-basic tree U such that $U_0 \cong F_0$ and $U' \cong C$.*

Proof. For future convenience, we modify outcomes described in subsection 3.3 by splitting them further. We have (e, x, ∞_i) indicating that there has been i stages at which some $\langle e', x'\rangle < \langle e, x\rangle$ increased its interval or moved/newly introduced its l-label. Similarly, we have (e, x, ∞_i), where i indicates that the predicate "fired" i times for smaller pairs. This modification will allow us to permamentrly abandon certain blocks in the construction.

We construct a computable group U represented by a computable p-basic tree. The group will be of the form

$$U = F \oplus \bigoplus_{\alpha} H(\alpha),$$

where α ranges over all outcomes of the procedure avoiding R, and F is the group (p-basic tree) of Ulm type at most 1 given by Fact 3.3. If α is the true outcome of the procedure then $H(\alpha)' \cong C$, and $H(\alpha)''' = 0$ otherwise. Additionally, $\bigcup_\alpha \#H(\alpha)_0 \subseteq \#F_0$, whence the diagonalization against R will be successful. It is sufficient to uniformly and independently construct the trees $H(\alpha)$ for different α. We explain the case $\alpha \neq g$ first, and then describe what happens if $\alpha = g$.

The case of $\alpha \neq g$. Notice that all outcomes $\alpha \neq g$ have the property that the whole $H(\alpha)$ is either $\mathbf{0}$ or will be forever abandoned if α is not the true outcome. If $\alpha = (e, n, \infty_i)$ for some i, then it will be using only lengths that are too small compared to $l_{e,x,s}$ a stage s. It will additionally make sure that lengths $l_{e',x',s}$ for $\langle e', x'\rangle < \langle e, x\rangle$ are not present in $\#H(\alpha)[s]$, at every s. Similarly, if $\alpha = (e, n, k_i)$

of $\alpha = (e, n, f_i)$, we make progress in approximating C but not using chains of lengths $l_{e',x',s}$ for $\langle e', x'\rangle \leq \langle e, x\rangle$. The rest is the same as in Proposition 3.4.

If one of the intervals $I_{e',x'}$ ever increase, or one of the $l_{e',x'}$ ever is assigned to a new number, or if the current guess on $l_{e,x}$ was finitary and now changed, then the whole $H(\alpha)$ will be permanently abandoned. We will then follow the modified basic strategy (subsection 3.3) on the finite tree that α left behind. As it is readily seen, if α is the true outcome, then $H(\alpha)' \cong C$, and $H(\alpha)'' = \mathbf{0}$ otherwise. In both cases clearly $\#H(\alpha)_0 \subseteq \#F_0$.

The Π^0_3-outcome g and $H(g)$. The procedure constructing $H(g)$ will be working within a copy of $\omega^{<\omega}$ which may be viewed as a complete ω-branching tree with root $\emptyset$ growing downwards. It will be enumerating a sub-tree of $\omega^{<\omega}$. At the end, the Π^0_2-tree C will be imaged into $\omega^{<\omega}$ so that nodes of depth n in C are mapped to nodes of depth n in $\omega^{<\omega}$(i.e., level-by-level).

The procedure believes that all sizes except for the ones in $Y = \{l_{e,x,s} : e, x \leq s\}$ are available, and it will be using chains of lengths not in Y, unless Y changes. It will add finite chains to elements in $\omega^{<\omega}$ currently corresponding to C. If an element is indeed in C then it will be put into $H(\alpha)'$ in the limit. Recall that $l_{e,x,s} \leq l_{e,x,t}$ for $t \geq s$.

Construction At a stage s, only one of the three actions can be performed:

(1) *Approximating C*. In this case we may assume $I_{e,x}$ and $l_{e,x}$ are stable, for $\langle e, x\rangle \leq s$. We basically follow the main strategy suggested in Proposition 3.4 (more precisely, in case when $\alpha \neq g$) since we know which sizes can be used. Namely, we use chains of lengths $\leq n_{e,x,s}$ not in $\bigcup_{\langle e,x\rangle\leq s} I_{e,x,s}$ at stage s (recall that we have reserved plenty of sizes in-between the intervals). We also ensure that if a new simple chain of length y is added below a node σ, then
   (i.) $y + i \notin \bigcup_{\langle e,x\rangle\leq s} I_{e,x,s}$ and $y + i \leq n_{e,x,s}$ for each $i \leq |\sigma|$;
   (ii.) y is larger than the maximal length of chains already extending σ, if there are any.

(2) *One of the $I_{e,x}$ grows.* In this case $I_{e,x}$ increases in size, and all substrategies of smaller priority lift their intervals to large fresh numbers. For every $\sigma \in \omega^{<\omega}$ such that $|\sigma| = \langle e, x\rangle$, permanently abandon any τ extending σ which has been used by the construction. Approximating C will now proceed under σ within the segment of the Baire space extending σ *disjoint* from all such τ's. (Note: The only reason we may again visit τ or its extension is due to $l_{e',x'}$ increasing for some $\langle e', x'\rangle \leq \langle e, x\rangle$, since all other l-labels will be too large.)

(3) *One of the $l_{e,x}$ grows.* Then we follow the generalized basic strategy (subsection 3.2) possibly adding further splittings to chains that could potentially represent size $l_{e,x,s}$. We add a finite chain to a predecessor of τ if there exists a scenario in the construction in which τ could potentially represent size $l_{e,x,s}$, these include only finitely many options, since there are only finitely many initial segments of τ. We use only simple chains of sizes $< l_{e,x,s}$ not in $\{l_{e',x',s} : \langle e', x'\rangle < \langle e, x\rangle\}$.

Verification for $H(g)$. There are two cases in which a segment of the tree built by the procedure can be *abandoned* by the construction. The first case is when the Π^0_2-predicate representing C is eventually silent on input a. The finite subtree extending $\sigma \in \omega^{<\omega}$ associated to a may never be visited again for the sake of approximating C. The second case corresponds to one of the $I_{e,x}$ increasing in size, in which all subtrees built by the strategy and rooted at level $\langle e, x\rangle + 1$ of $\omega^{<\omega}$ will never be active for the sake of C-approximation. The fundamental difference is that the former is a Σ^0_2-event while the latter is Σ^0_1. We may still visit a forever-abandoned segment of the tree due to the l-labels moving, but not for the sake of approximating C.

Remark 3.6. Suppose T is a subtree which looks abandoned (Σ^0_2) or is permanently abandoned (Σ^0_1). It could have happened that the finite lengths which occur in the tree (after stripping) are not allowed. For instance, in [2, 31] (see Lemma 2.7) one has to extend the longest chain present in T carefully using the limitwise monotonic function, and then observe the construction. We do not have to do that in our construction because of (i.) and (ii.) above, unless one of the sufficiently small l-labels moves to a larger number.

The construction is organized so that are only two cases at which we might have to act due to $l_{e,x}$ increasing:

Case 1 We permanently left behind a finite tree T due to $I_{e,x}$ or some other interval increasing. Starting from this step on, we follow the generalized basic strategy (subsection 3.2) in our actions on this finite tree. It will never be used to approximate C again.

Case 2 We have left behind a finite tree due to C being silent on one of its inputs, say on c. The node σ currently representing c in Baire space may have arbitrary long finite chains attached to it (it includes the case when $|\sigma| < \langle e, x\rangle$). We follow the generalized basic strategy (subsection 3.2) on each of the chains, thus possibly further branching some of the chains extending σ. Note that σ may be visited again due to a new C-activity on c.

In Case 1, the are only finitely many l-labels which are small enough and can potentially force us to add new simple chains to T. For simplicity, suppose there are only three of them, l_0, l_1 and l_2. The label l_0 can possibly move only up to the end of its interval. If its interval ever increases, both l_1 and l_2 will be lifted large. The we can argue as in subsection 3.2. Based on this idea, we prove:

Claim 3.7. *In Case 1, tree-ranks of nodes in T are bounded by $\omega + k$ at every stage of the construction, for some fixed $k \in \omega$ not depending on the stage (but depending on the tree).*

Proof. Suppose T is abandoned at stage s. There are only finitely many l-labels which can potentially increase the rank of a node $\sigma \in T$. If all of these labels are eventually stable or too large, the rank of $\sigma \in T$ is finite in the limit. Let l_i be the least among these labels which tends to infinity. It follows that the corresponding interval containing l_i has to be increased at a stage $t > s$. All labels l_j with $j > i$ are lifted up to large numbers at stage t, and they can not effect the ranks of nodes in T anymore. Let $t' \geq t$ be a stage after which all labels less than l_j are stable. Let k be the maximal length in the tree T', where T' consists of T and all chains added for the sake of avoiding l-labels at stages $\leq t'$. Since from stage t' we follow the modified diagonalization strategy (subsection 3.2), the ranks of predecessors of terminal nodes in T' are at most ω. □

In Case 2, the worst scenario is when σ representing c is of length smaller than the least $\langle e,x\rangle$ for which $I_{e,x}$ is grows to infinity. Then C may fire on c in-between $I_{e,x}$-expansionary stages, and longer simple chains will be added to σ. Then these chains will become infinitely branching due to $l_{e,x,s}$ increasing. In this case we end up with σ having rank $\omega\cdot 2+k$, for some $k\leq\langle e,x\rangle$. We summarize these ideas in the claim below:

Claim 3.8. *Suppose the true outcome of the main R-avoiding procedure is not $\boldsymbol{g}$. Then tree-ranks of nodes in $H(g)$ are bounded by $\omega\cdot 2+k$, for some fixed $k\in\omega$.*

Proof. Suppose the true outcome is of the form $(e,x,\cdot)$. It corresponds to the case when every σ having length $\geq\langle e,x\rangle+1$ ever introduced by the construction is permanently abandoned at some stage. Then every σ exceeding $\geq\langle e,x\rangle+1$ in length may have rank at most $\omega+k$, for some k. Whence, every node in $\omega^{<\omega}\restriction\langle e,x\rangle+1$ has successors of ranks bounded by $\omega\cdot 2$. Therefore, the rank of the root $\emptyset$ is at most $\omega\cdot 2+\langle e,x\rangle+1$. □

Finally, we prove:

Claim 3.9. *If $\boldsymbol{g}$ is the true outcome of the R-avoiding procedure, then $H(g)'\cong C$.*

Proof. Observe that the only reason a permanently abandoned node (Case 1) may have an infinite rank is when one of the l-labels tend to infinity. Therefore, all permanently abandoned nodes may contribute only to $H(g)_0$. The same argument applies if a node is abandoned due to C being eventually silent on the corresponding input (Case 2). We need to verify that ranks of nodes in the simple chains added to a node σ for the sake of approximating C will be kept finite in the limit. Note that the simple chain can be further branched, using chains of smaller length, due to one of sufficiently small l-labels moving. Only finitely many labels may force us to further branch the simple chain, all other labels will occupy numbers which are too large. All labels, if defined, have to be eventually settled at finite locations. Thus, we may potentially end up with a finite tree properly containing the original simple chain. It is clear that σ will have rank at least $\omega+|\sigma|$ if $c\in C$. It may have a larger rank only if there exists c' extending c in C. □

We have verified that $H(g)'\cong C$ if g is the true outcome, and $H(g)'''=0$, otherwise. It is also clear from the construction that in both cases $\#H(g)_0\subseteq\#F_0$. It completes the verification for $H(g)$. □

3.7. **Finalizing the proof.** Using Fact 3.3, we can produce a uniformly $\Pi^0_{(6e)}$-sequence $(A(3e))_{e\in\omega}$ of $R_{e,3e}$-avoiding p-basic trees. By Propositions 3.4 and 3.5, there exist a uniformly computable sequence $(B(i))_{i\in\omega}$ of computable p-basic trees such that, for each $k\leq i$, $B(i)_k\cong A(k)$ if $k=3e$, and $B(i)_k\cong\bigoplus_{m,n}Z_{p^m}a_{m,n}$ otherwise.

We set G equal to $\bigoplus_{i\in\omega}B(i)$. By the definition of G, and since $A(3e)\cong A(3e)\oplus A(3e)$ for every e, the requirement L_e is met for each e. Since the operation of taking a direct sum (of p-basic trees, defined naturally) is uniform, and the p-basic trees $(B(i))_{i\in\omega}$ are computable uniformly in i, the p-basic tree for G is computable.

4. An application of p-basic trees to categoricity

In this section, we use the machinery of p-basic trees to prove:

Theorem 1.8. Let G be a computable p-group of finite Ulm type n, such that:
(1.) $G^{(n)} \cong \oplus_{j \leq m} Z_{p^\infty}$, for some $m \in \omega$;
(2.) orders of cyclic summands in G_{n-1} are not bounded.
Then G is not Δ^0_{2n}-categorical.

Proof. The group G can be represented as

$$G = H \oplus D,$$

where $D \cong \oplus_{j \leq m} Z_{p^\infty}$, and H is reduced of type n. In this proof, "tree" is used in several meanings, but it should be clear from the context what exactly we mean; also these meanings are (effectively) interchangeable. We need:

Fact 4.1 (Folklore)**.** *G has a computable copy if, and only if, H has a computable copy.*

Proof sketch. The right-to-left implication is elementary. For the left-to-right implication, non-uniformly fix the finite subspace of the socle of G_k, $k \leq n$, consisting of elements of H. Then apply Define a monotonic function approximating heights of the remaining elements in G_k. Then apply Theorem 1.4 to H. □

Encode Tot as a Π^0_2 path through $\omega^{<\omega}$. Denote the resulting Π^0_2-tree (a singe path) by P. The coding of Tot and the fact above can be relativized to a coding of a Π^0_{2n}-complete set into elements of rank ∞ of a Δ^0_{2n-1}-tree. We can define a Δ^0_{2n-1}-isomorphism of this Δ^0_{2n-1}-tree onto a Π^0_{2n-2} subtree of $\omega^{<\omega}$, and then apply Lemma 2.7 to expand the resulting Π^0_{2n-2} to a Δ^0_{2n-3}-tree. We repeat until we get a *computable* p-basic tree S. Note that the embeddings that we used were at most Δ^0_{2n-1} or less complex.

We can effectively adjoin $m-1$ infinite chains to the root of S and obtain a p-basic tree representing a computable copy A of G. We can also take a computable copy of H (represented by a p-basic tree) and adjoin m infinite chains to its root. Let B denote the resulting group (tree).

If A and B are isomorphic via a Δ^0_{2n}-isomorphism f, then we can reconstruct the Π^0_{2n}-complete set encoded into A considering images of $a \in A$ in the group B (in which the divisible part is a computable subgroup). □

5. Conclusion

We have shown that guessing limitwise monotonic functions at appropriate level of a group of type ω is as hard as they could be. We expect that Theorem 1.5 can be pushed to any computable ordinal.

We note that the group constructed in Theorem 1.5 has a complex uniformity property, but we circumvented many algebraic difficulties by specifically choosing Ulm invariants (homogenizing G_{3e+1} and G_{3e+2}) and their representations (Note 3.2, Remark 3.6). It also seems crucial for the construction that the l-labels can only be moved to larger numbers. Dropping at least one of these restrictions would result serious problems such as a simultaneous interaction of infinitely many strategies. It is not surprising that the classification of p-groups of Ulm type ω is a largely unexplored area.

Let Γ be either Π or Σ or Δ, and let $m \leq 3$. Consider the class $\mathcal{R}_p(\Gamma, m)$ of computable reduced p-groups A of Ulm type ω in which $0^{(2n)}$-indices for limitwise monotonic functions ranging over $\#A_n$ are uniformly Γ_{n+m}.

It is readily checked that

$$\mathcal{R}_p(\Delta,1)=\mathcal{R}_p(\Delta,2),$$

methods developed in our paper can be applied to show

$$\mathcal{R}_p(\Delta,2)\subsetneq\mathcal{R}_p(\Pi,2),$$

and Theorem 1.5 implies

$$\mathcal{R}_p(\Delta,3)\subsetneq\mathcal{R}_p(\Pi,3).$$

What can be said about the other inclusions?

The following result is well-known:

Proposition 5.1. [2] *Groups in $\mathcal{R}_p(\Delta,1)$ are in $1-1$ correspondence with uniformly $0^{(2n+1)}$-limitwise monotonic collections of Ulm invariants.*

Proof idea. We can split each limitwise monotonic set into infinitely many infinite disjoint subsets. We then uniformly run the proof of Theorem 1.4 and obtain a uniform sequence of computable groups $(H_n)_{n\in\omega}$, using more of the limitwise monotonic disjoint subsets for larger n, and so that all disjoint subsets are used in one of the H_n. We then pass to $\bigoplus_{n\in\omega} H_n$ which has the desired invariants. □

Problem 5.2. *Prove an analog of Proposition 5.1 for $\mathcal{R}_p(\Pi,2)$.*

The first case to consider would be groups G having $\#G_i$ co-finite, or even $\#G_i$ either ω or a co-singleton. There are some purely algebraic obstacles even in this simplest case. It may be the case that not every infinite sequence of $0^{(2n+1)}$-limitwise monotonic infinite sets corresponds to a computable reduced p-group. If there are any further necessary conditions, they might become visible already in $\mathcal{R}_p(\Pi,2)$.

We leave open:

Problem 5.3 (Ash, Knight, Oates)**.** *Does every computable p-group have a computable p-basic tree?*

References

[1] B. Andresen, A. Kach, A. Melnikov, and D. Solomon. Jump degrees of torsion-free abelian groups. To appear in Journal of Symbolic Logic.

[2] C. Ash, J. F. Knight, and S. Oates. Recursive abelian p-groups of small length. Unpublished. An annotated manuscript: `https://dl.dropbox.com/u/4752353/Homepage/AKO.pdf`.

[3] 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.

[4] Wesley Calvert. The isomorphism problem for computable abelian p-groups of bounded length. *J. Symbolic Logic*, 70(1):331–345, 2005.

[5] Wesley Calvert, Douglas Cenzer, Valentina S. Harizanov, and Andrei Morozov. Effective categoricity of abelian p-groups. *Ann. Pure Appl. Logic*, 159(1-2):187–197, 2009.

[6] Richard J. Coles, Rod Downey, and Bakhadyr Khoussainov. On initial segments of computable linear orders. *Order*, 14(2):107–124, 1997/98.

[7] Barbara F. Csima, Denis R. Hirschfeldt, Julia F. Knight, and Robert I. Soare. Bounding prime models. *J. Symbolic Logic*, 69(4):1117–1142, 2004.

[8] J. C. E. Dekker. Countable vector spaces with recursive operations. I. *J. Symbolic Logic*, 34:363–387, 1969.

[9] J. C. E. Dekker. Countable vector spaces with recursive operations. II. *J. Symbolic Logic*, 36:477–493, 1971.

[10] R. Downey, A. Kach, and D. Turetsky. Limitwise monotonic functions and applications. In *Proceedings of STACS 2012*, pages 56–85, 2011.

[11] R. Downey and A. G. Melnikov. Effectively categorical abelian groups. To appear in Journal of Algebra.

[12] R. G. Downey. Computability theory and linear orderings. In *Handbook of recursive mathematics, Vol. 2*, volume 139 of *Stud. Logic Found. Math.*, pages 823–976. North-Holland, Amsterdam, 1998.

[13] R. G. Downey and Stuart A. Kurtz. Recursion theory and ordered groups. *Ann. Pure Appl. Logic*, 32(2):137–151, 1986.

[14] Rodney G. Downey. On presentations of algebraic structures. In *Complexity, logic, and recursion theory*, volume 187 of *Lecture Notes in Pure and Appl. Math.*, pages 157–205. Dekker, New York, 1997.

[15] Yuri L. Ershov and Sergei S. Goncharov. *Constructive models*. Siberian School of Algebra and Logic. Consultants Bureau, New York, 2000.

[16] A. Fröhlich and J. C. Shepherdson. Effective procedures in field theory. *Philos. Trans. Roy. Soc. London. Ser. A.*, 248:407–432, 1956.

[17] László Fuchs. *Infinite abelian groups. Vol. I.* Pure and Applied Mathematics, Vol. 36. Academic Press, New York, 1970.

[18] László Fuchs. *Infinite abelian groups. Vol. II.* Academic Press, New York, 1973. Pure and Applied Mathematics. Vol. 36-II.

[19] Sergei S. Goncharov. *Countable Boolean algebras and decidability*. Siberian School of Algebra and Logic. Consultants Bureau, New York, 1997.

[20] Sergey S. Goncharov, Steffen Lempp, and Reed Solomon. The computable dimension of ordered abelian groups. *Adv. Math.*, 175(1):102–143, 2003.

[21] Kenneth Harris. η-representation of sets and degrees. *J. Symbolic Logic*, 73(4):1097–1121, 2008.

[22] Denis Hirschfeldt, Russell Miller, and Sergei Podzorov. Order-computable sets. *Notre Dame J. Formal Logic*, 48(3):317–347, 2007.

[23] Denis R. Hirschfeldt. Prime models of theories of computable linear orderings. *Proc. Amer. Math. Soc.*, 129(10):3079–3083 (electronic), 2001.

[24] Denis R. Hirschfeldt, Bakhadyr Khoussainov, and Pavel Semukhin. An uncountably categorical theory whose only computably presentable model is saturated. *Notre Dame J. Formal Logic*, 47(1):63–71 (electronic), 2006.

[25] N. G. Hisamiev. The periodic part of a strongly constructivizable abelian group. In *Theoretical and applied problems in mathematics and mechanics (Russian)*, pages 299–303, 318. "Nauka" Kazakh. SSR, Alma, 1977.

[26] N. G. Hisamiev. Criterion for constructivizability of a direct sum of cyclic p-groups. *Izv. Akad. Nauk Kazakh. SSR Ser. Fiz.-Mat.*, (1):51–55, 86, 1981.

[27] Asher M. Kach. Computable shuffle sums of ordinals. *Arch. Math. Logic*, 47(3):211–219, 2008.

[28] I. Kalimullin, B. Khoussainov, and A. Melnikov. Limitwise monotonic sequences and degree spectra of structures. To appear.

[29] Irving Kaplansky. *Infinite abelian groups*. Revised edition. The University of Michigan Press, Ann Arbor, Mich., 1969.

[30] N. G. Khisamiev. Hierarchies of torsion-free abelian groups. *Algebra i Logika*, 25(2):205–226, 244, 1986.

[31] N. G. Khisamiev. Constructive abelian p-groups. *Siberian Adv. Math.*, 2(2):68–113, 1992. Siberian Advances in Mathematics.

[32] 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.

[33] 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.

[34] Charlotte Lin. Recursively presented abelian groups: effective p-group theory. I. *J. Symbolic Logic*, 46(3):617–624, 1981.

[35] A. I. Mal′cev. On recursive Abelian groups. *Dokl. Akad. Nauk SSSR*, 146:1009–1012, 1962.

[36] G. Metakides and A. Nerode. Recursively enumerable vector spaces. *Ann. Math. Logic*, 11(2):147–171, 1977.

[37] G. Metakides and A. Nerode. Effective content of field theory. *Ann. Math. Logic*, 17(3):289–320, 1979.

[38] A.T. Nurtazin. *Computable classes and algebraic criteria of autostability*. Summary of Scientific Schools, Math. Inst. SB USSRAS, Novosibirsk, 1974.

[39] Sarah Elizabeth Oates. *Jump degrees of groups*. ProQuest LLC, Ann Arbor, MI, 1989. Thesis (Ph.D.)–University of Notre Dame.

[40] Michael O. Rabin. Computable algebra, general theory and theory of computable fields. *Trans. Amer. Math. Soc.*, 95:341–360, 1960.

[41] J. B. Remmel. Recursive Boolean algebras. In *Handbook of Boolean algebras, Vol. 3*, pages 1097–1165. North-Holland, Amsterdam, 1989.

[42] Laurel A. Rogers. Ulm's theorem for partially ordered structures related to simply presented abelian p-groups. *Trans. Amer. Math. Soc.*, 227:333–343, 1977.

[43] Rick L. Smith. Two theorems on autostability in p-groups. 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 302–311. Springer, Berlin, 1981.

[44] Robert I. Soare. *Recursively enumerable sets and degrees*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.