EFFECTIVELY CLOSED SUBGROUPS OF THE INFINITE SYMMETRIC GROUP

NOAM GREENBERG, ALEXANDER MELNIKOV, ANDRE NIES, AND DANIEL TURETSKY

ABSTRACT. We apply methods of computable structure theory [AK00, EG00] to study effectively closed subgroups of S_∞. The main result of the paper says that there exists an effectively closed presentation of $\mathbb{Z}_2$ which is not the automorphism group of any computable structure M. In contrast, we show that every effectively closed discrete group is topologically isomorphic to $\mathrm{Aut}(M)$ for some computable structure M. We also prove that there exists an effectively closed compact (thus, profinite) subgroup of S_∞ that has no computable Polish presentation. In contrast, every profinite computable Polish group is topologically isomorphic to an effectively closed subgroup of S_∞. We also look at oligomorphic subgroups of S_∞; we construct a Σ^1_1 oligomorphic group in which the orbit equivalence relation is not uniformly HYP. Our proofs rely on methods of computable analysis, techniques of computable structure theory, elements of higher recursion theory, and the priority method.

The study of computable presentations of topological groups originated in computable field theory [MN79] and was mainly driven by Nerode's interest in algorithmic aspects of Krull theory. Working under the supervision of Nerode, La Roche [LR81] proved that the correspondence between computable algebraic number field extensions and profinite groups is uniformly effective, in the sense that will be clarified later. Quite interestingly, the algorithmic techniques developed in [LR81] allowed La Roche to prove a theorem on free profinite groups that was new even in the purely algebraic (non-computable) setting, see [Jar74] for the earlier and a weaker purely algebraic result. Based on the work of La Roche, Smith [Smi81, Smi79] initiated the study of algorithmic presentations of profinite groups in their own right, i.e. not in the context of effective Galois theory.

Such investigations in computable topological groups have not been restricted to profinite groups (see., e.g., [GR93]), but the general theory of computable Polish groups is still in its infancy. Recently there has been an increasing interest in computable aspects of Polish and Banach spaces [PER89, BHW08, Mel13, McN15] and, consequently, in computable Polish groups [MM, Mel]. Many aspects of computable Polish groups are related to computable structure theory [AK00, EG00] and computable Banach space theory [PER89]. Such connections are often quite subtle. For example, it turns out that many classical results of computable structure theory have simpler proofs in the more general setting of a computable Polish group action, see [MM]. On the other hand, the study of Pontryagin Duals of computable Polish abelian groups enjoys applications of non-trivial effective algebraic results, see [Mel]. It seems that effective algebra and computable topological group theory are two adjacent pieces of a bigger puzzle. This paper contributes to the general framework proposed in [Mel13] that is focused on establishing further technical connections between computable structure theory and computable analysis, see also [MN16, MN13, MM, GMKT, McN15, MS, Mel].

Recall that a countably infinite and discrete algebraic structure (e.g., a countable field of characteristic 0) is *computable* if its domain is ω and its operations and relations are Turing

The first author was partially supported by the Marsden Fund of New Zealand. The second author was partially supported by the Marsden Fund of New Zealand and Massey University Research Fund.

computable. Our main goal is to investigate automorphism groups of computable algebraic structures. For this purpose we introduce a new notion of an *effectively closed group.* (Our second main result will imply that the notion below is indeed new.)

Definition 0.1. We say that a subgroup G of S_∞ is *effectively closed* (or Π^0_1 for short) if there is an effectively closed subset P of ω^ω such that $G = P \cap S_\infty$. An *effectively closed presentation*, or a Π^0_1*-presentation*, of a group is an effectively closed subgroup of S_∞ topologically isomorphic to the group.

Automorphism groups of computable algebraic structures are clearly effectively closed. It is well-known that every closed subgroup of S_∞ is equal to the automorphism group of some countable algebraic structure upon ω, see [Gao09]. It is natural to ask:

Is every effectively closed group equal to $\mathrm{Aut}(\mathcal{C})$ for some computable $\mathcal{C}$?

We will see that the answer to this question is negative, which seems somewhat counterintuitive. The reader perhaps suspects that the isomorphism type of any effectively closed subgroup of S_∞ witnessing the negative answer should be, in some sense, non-trivial. Remarkably, already the two-element cyclic discrete group $\mathbb{Z}_2$ has a "bad" effectively closed presentation. On the other hand, $\mathbb{Z}_2$ is (topologically) isomorphic to $\mathrm{Aut}(\mathcal{C})$ for some computable $\mathcal{C}$.

Theorem 0.2.

(1) There exists an effectively closed presentation of the two-element cyclic group $\mathbb{Z}_2$ such that $G \neq \mathrm{Aut}(\mathcal{C})$ for any computable structure $\mathcal{C}$.
(2) Every effectively closed discrete group is topologically isomorphic to $\mathrm{Aut}(\mathcal{C})$ for some computable structure $\mathcal{C}$.

The main difficulty in the proof of Theorem 0.2(1) is nesting strategies on top of each other and not losing the property of being a subgroup of S_∞. We leave open whether Theorem 0.2(2) can be extended to non-compact effectively closed groups.

One could also argue that Definition 0.1 is natural in its own right; effectively closed sets play a significant role in recursion theory, for some recent applications see [BC08, Rei08, HK14]. It could be the case that the notion is actually equivalent to one of the already existing notions restricted to closed subgroups of S_∞. We compare Definition 0.1 with the mentioned above notions of a *computable Polish group* [MM, Mel] and a *recursive profinite group* [LR81, Smi81]. (The formal definitions will be given in the preliminaries.) Every recursive profinite group is computable Polish, but there are computable Polish profinite groups with no recursive presentation [Mel]. Recall that the profinite groups are exactly the compact subgroups of S_∞, up to topological isomorphism.

Theorem 0.3.

(1) There exists an effectively closed compact (thus, profinite) subgroup of S_∞ that has no computable Polish presentation (therefore, no recursive presentation).
(2) Every profinite computable Polish group is topologically isomorphic to an effectively closed subgroup of S_∞.

In particular, part (1.) of Theorem 0.3 shows that Definition 0.1 gives a new notion, while part (2.) establishes a connection between Definition 0.1 and computable Polish groups. In the preliminaries we will use similar ideas to suggest an extension of Definition 0.1 to groups that are not necessarily totally disconnected. We leave open whether Theorem 0.3(2) can be extended to arbitrary closed subgroups of S_∞.

To finish the paper we look at the class of oligomorphic groups. There are closed subgroups G of S_∞ for which for every n there are only finitely many G-orbit equivalence classes of n-tuples. Oligomorphic groups are the automorphism groups of $\aleph_0$-categorical structures. These structures are homogenous. Thus, if an effectively closed subgroup G of S_∞ is oligomorphic, and equals $\mathrm{Aut}(M)$ for some computable structure M, then $\sim_G$ will in fact be *computable*. We approach the question "how complicated can $\sim_G$ be for an effectively closed oligomorphoc G?" If we could construct an effectively closed oigomorphic group with $\sim_G$ not computable, we would get another example for our main result Theorem 0.2(1). Oligomorphic groups lie at the other extreme from profinite groups, for which every orbit equivalence class is finite, and our initial hope was that it might perhaps be easier to handle them. Our intuition was wrong, but the effective content of oligomorphic groups turned out to be interesting on its own right.

Note that even for Σ^1_1 groups, $\sim_G$ is Σ^1_1; if G is oligomorphic, then for each n, the restriction of $\sim_G$ to n-tuples must be hyperarithmetic. Interetingly enough, this fact lacks uniformity in the following sense.

Theorem 0.4. There is a Σ^1_1, closed oligomorphic subgroup of S_∞ for which $\sim_G$ is not hyperarithmetic.

It would be interesting to obtain more information about the effective content of oligomorphic groups; in particular, we leave open whether a Π^0_1 oligomorphic group can witness Theorem 0.2(1).

The structure of this paper is as follows. The small preliminary Section 1 contains formal definitions, a description of the natural computable Polish presentation of S_∞, and an equivalent definition of an effectively closed subgroup of S_∞ in terms of this presentation. We arrange proofs according to the used methods. The proof of Theorem 0.3 can be found in Section 2. Section 3 contains the proof of Theorem 0.2(1), and the proofs of Theorem 0.2(2) and Theorem 0.4 are in Section 4, respectively.

1. Preliminaries

Throughout this paper we work in the category of topological groups. We only consider isomorphisms between groups that are both algebraic and topological, i.e., homeomorphisms; so henceforth "isomorphic" means "topologically isomorphic".

Definition 1.1. A *computable Polish (metric) space* is a triple $(M, d, (\alpha_i)_{i\in\omega})$, where (M, d) is a Polish space, the sequence $(\alpha_i)_{i\in\omega}$ is dense in M, and there exists a uniformly computable procedure which on input $i, j \in \omega$ and $\epsilon \in \mathbb{Q}^+$, outputs a rational r such that $|d(\alpha_i, \alpha_j) - r| < \epsilon$.

The points from the dense computable sequence $(\alpha_i)_{i\in\omega}$ are called *special*. A ball with a rational radius and centred in a special point is called *basic*. Clearly, ω^ω under the usual ultrametric forms a computable Polish space. The basic open balls are the usual clopen neighbourhoods.

Suppose $F : \mathbb{X} \to \mathbb{Y}$ is a map between computable Polish spaces. The map F is *computable* if there exists a uniform procedure which, on input a basic open $B \subset \mathbb{Y}$, lists a set of basic open balls in $\mathbb{X}$ whose union is equal to $F^{-1}(B)$. Note that we do not require to list all balls contained in $F^{-1}(B)$. This approach is equivalent to the definition from, e.g., [Mel13]. Observe that the product of two (or more) computable Polish spaces is itself computable Polish. The definition below generalises the old definition of a recursive profinite group [LR81, Smi81] to arbitrary Polish groups.

Definition 1.2 ([MM]). A *computable Polish presentation* of a [second-countable Hausdorff] topological group G is a computably and completely metrized homeomorphic copy of G upon which the operations $\cdot$ and $^{-1}$ are computable.

We now discuss the infinite permutation group $S_\infty \subset \omega^\omega$. For two permutations σ and τ of ω, let

$$D(\sigma,\tau) = \frac{d(\sigma,\tau) + d(\sigma^{-1},\tau^{-1})}{2},$$

where d is the usual ultrametic on ω^ω. The dense computable subset of S_∞ is given by permutations of ω with finite support. Then (S_∞, D) becomes a computable Polish group which we call the natural Polish presentation of S_∞ for reasons that will be explained shortly.

Recall that a closed subset C of a computable Polish space is *effectively closed* or Π^0_1 if there exists a (Turing) computable enumeration of basic open balls whose union is the complement of C, and recall Definition 0.1 of an effectively closed subgroup of S_∞. It seems quite natural and certainly useful to define a Π^0_1-subgroup based on the natural computable Polish presentation of S_∞ of ω^ω instead. Luckily these notions coincide.

Proposition 1.3. A group $G \leqq S_\infty$ is effectively closed (according to Definition 0.1) if and only if the domain of G is an effectively closed subset of the natural computable Polish presentation of S_∞.

Proof. We prove a bit more. Suppose that U is a basic open subset of ω^ω. Can we effectively list basic open subsets of (S_∞, D) whose union is equal to $S_\infty \cap U$? Conversely, given a basic open V in (S_∞, D), can we list basic open sets whose union U has the property $U \cap S_\infty = V$? In other words, is the metric D (on S_∞) effectively compatible with the usual metric d (on ω^ω)?

Claim 1.4. The metric D defined above is effectively compatible with the standard ultrametric d on ω^ω.

Proof of Claim. Suppose that a sub-basic open subset U of ω^ω is specified by a finite partial map σ. If σ is not injective, then $V = \emptyset$. Otherwise, let τ_1, τ_2, ... be the effective list of all possible extensions of σ^{-1} to an injective map with domain a finite initial segment of ω. Then the sequence of pairs (τ_i^{-1}, τ_i), $i = 1, 2, \ldots$ corrsponds to a sequence of respective basic open balls in (S_∞, D) whose union we set equal to V. Note that each element of $U \cap S_\infty$ belongs to one of these basic open sets, and thus to V. Conversely, each $f \in V$ has a finite approximation that is consistent with the restriction imposed by U.

Now suppose (σ, σ^{-1}) describes a basic open set V in (S_∞, D), where σ is a permutation of an initial segment of ω. Then the open set U can be taken equal to the collection of all extensions of σ to an element of ω^ω. The correspondence is clearly effective. □

Let P be a closed subset of ω^ω. It follows that we can uniformly pass from the enumeration of $\overline{P}$ in ω^ω to the enumeration of $\overline{P} \cap S_\infty$ in (S_∞, D) and back. □

The proposition above allows us to extend Definition 0.1 to groups that are not necessarily embeddable into S_∞, but we leave this new notion outside the scope of this article.

Definition 1.5 ([LR81, Smi81]). A *recursive presentation* of a profinite group P is a uniformly computable inverse system of finite groups, with surjective maps, whose inverse limit is isomorphic to P.

It is known that recursively presented profinite groups are exactly the automorphism groups of computable algebraic number fields over a computable subfield; see [LR81]. As we mentioned above, every recursive profinite group is computable Polish, but there are computable Polish profinite groups with no recursive presentation [Mel].

2. Effectively closed vs. computable Polish

Proof of Theorem 0.3(1). We construct an effectively closed gubgroup of S_∞ that has no computable Polish presentation. We will be using the result below:

Fact 2.1 ([Mel], Cor. 1.8)**.** Every computable Polish presentation of a profinite group P can be transformed into a $0'$-computable inverse system $F_0 \leftarrow F_1 \leftarrow F_2 \ldots$, with surjective maps, whose limit is isomorphic to P.

For a set S of prime numbers let

$$P_S = \prod_{p \in S} \mathbb{Z}_p,$$

where $\mathbb{Z}_p$ is the cyclic group of order p. This group is profinite, as it is the inverse limit of the groups $\prod_{p \in S \cap n} \mathbb{Z}_p$ for $n < \omega$.

First we observe that if P_S has a computable Polish presentation then S is Σ^0_2. To see this, given such a presentation of P_S, by Fact 2.1, we produce a $0'$-computable inverse system representing the group. By [Mel, Thm.1.9] we can use this to uniformly produce a $0'$-computable presentation – in the sense of computable structure theory – of the discrete countable group $\bigoplus_{p \in S} \mathbb{Z}_p$, which is the *Pontryagin dual* of P (see the book [Pon66] for more on Pontryagin's duality theory). Using the $0'$-computable presentation of $\bigoplus_{p \in S} \mathbb{Z}_p$ we can $0'$-computably list the prime orders of elements of the group, showing that S is Σ^0_2.

So it suffices, given a Π^0_2-complete set S of primes, to build a computable structure M such that $\mathrm{Aut}(M)$ is isomorphic to P_S. The structure M will be a graph consisting of infinitely many disjoint components C_p, one for each prime p. Every C_p will have a loop of length p,

$$x^p_0 - x^p_1 - \ldots - x^p_{p-1} - x^p_0$$

and every node in the loop will have a [finite or infinite] chain

$$x^p_i - c^p_{i,1} - c^p_{i,2} - \ldots$$

attached to it. The length of the chain depends on our approximation for the Π^0_2 predicate for p. If $p \notin S$ then this predicate fires for p only finitely many times, say s; in this case, we make the length of the ith chain equal to $s + i$. The result is a rigid component. If $p \in S$ then we make each of the p many chains infinite. In this case the automorphism group of the component will be isomorphic to $\mathbb{Z}_p$; each automorphism is determined by the image of x^p_0, which could be any x^p_i.

Because there is no interaction between the components, $\mathrm{Aut}(M) \cong \prod_{p \in S} \mathrm{Aut}(C_p) \cong P_S$. This isomorphism is topological as well, because in both copies, the topology is the product topology where the components $\mathrm{Aut}(C_p)$ and $\mathbb{Z}_p$ are discrete. In other words, in both $\mathrm{Aut}(M)$ and C_p, the sub-basic clopen sets are determined by stating finitely many values for the automorphism. □

Proof of Theorem 0.3(2). Let P be a computable Polish profinite group. We need to produce a Π^0_1 presentation of P. By Fact 2.1, there is a $0'$-recursive presentation of P. We will use a fully relativised form of the fact below.

Fact 2.2 (LaRoche [LR81])**.** Every recursively presented profinite group is isomorphic to $\mathrm{Gal}(K/N)$, where K is a computably presented algebraic extension of $\mathbb{Q}$, and N is a computable subfield of K.

In fact, N is a fixed field that corresponds to a natural recursive presentation of the free profinite group upon countably many generators, see [LR81]; we do not need this fact.

Fix a $0'$-computable field K and a $0'$-computable subfield N of K such that $\mathrm{Aut}(K/N) \cong P$. Our first step is to obtain a $0'$-computable relational structure F (with computable underlying set and in a computable language) such that $\mathrm{Aut}(F) \cong \mathrm{Aut}(K/N)$ (all isomorphisms are topological); then we obtain a computable structure $\hat{F}$ such that $\mathrm{Aut}(\hat{F}) \cong \mathrm{Aut}(F)$.

To define F, we start with K; by taking a $0'$-computably isomorphic copy we may assume that N is computable. We name each elements of N by singleton unary predicate and replace the field operations by their graphs. This adjustment does not change the topological isomorphism type of the automorphism group, so $\mathrm{Aut}(F) \cong P$.

The next step is a version of Marker's existential extension which preserves the automorphism group.

Proposition 2.3. For any $\emptyset'$-computable relational structure A there is a computable structure B such that $\mathrm{Aut}(A)$ and $\mathrm{Aut}(B)$ are isomorphic.

Proof. We use the following piece of folklore in computable structure theory.

Fact 2.4 (Folklore, e.g., [GK02])**.** Let S be a Σ^0_2 set. There exists a uniform procedure which, for each $x \in \omega$, outputs a computable copy of ω if $x \in S$, and outputs a computable copy of ω^2 otherwise.

We assume that the underlying set of A is computable, with a computable language. To define B, fix an n-ary relation of A. For every tuple $\bar{a} \in A^n$ we add a new infinite set $C^P_{\bar{a}}$ of elements (these are pairwise disjoint). We link these "blow ups" of tuples by adding the relation $y \in C^P_{\bar{x}}$ to B. Next, for every tuple $\bar{a} \in A^n$ we define a linear ordering $L^P_{\bar{a}}$ of $C^P_{\bar{a}}$, which is isomorphic to ω if $P(\bar{a})$ holds in A, and isomorphic to ω^2 otherwise. We add the relation $y_1, y_2 \in C^P_{\bar{x}} \,\&\, y_1 <_{L^P_{\bar{x}}} y_2$ to the structre B. It follows from Fact 2.4 that B has a computable copy.

To show that $\mathrm{Aut}(A) \cong \mathrm{Aut}(B)$, we notice that both ω and ω^2 are rigid. Thinking of the underlying set of A as a (computable, definable) subset of B, we observe that every automorphism of B is determined by its restriction to A, and that every automorphism of A can be extended to an automorphism of B. Let $\Phi\colon \mathrm{Aut}(B) \to \mathrm{Aut}(A)$ be this restriction map, $\Phi(\tau) = \tau \restriction A$; it is the required isomorphism. To see that it is topological, in the slightly less immediate direction, we need to check that it is an open map. We take a sub-basic clopen subset of $\mathrm{Aut}(B)$, which is the collection of automorphisms of B which extend some finite map σ. The point is that σ may mention some elements of $B \setminus A$. Nonetheless, $\Phi[\sigma]$ is clopen in $\mathrm{Aut}(A)$. If $q \in C^P_{\bar{a}}$ is mapped to some $p \in C^P_{\bar{b}}$, then to the image of $[\sigma]$ we add the restriction that $\bar{a}$ is mapped to $\bar{b}$. Since ω and ω^* are rigid, mapping $\bar{a}$ to $\bar{b}$ is equivalent to mapping q to p. ☐

This completes the proof of Theorem 0.3(2). ☐

3. Proof of Theorem 0.2

We must construct a Π^0_1 presentation of $\mathbb{Z}_2$ which is not equal to $\mathrm{Aut}(M)$ for any computable structure M (upon the domain of ω).

Informal idea. We explain the basic idea behind diagonalising against the eth partial computable structure M_e. We work in ω^ω. We start by enumerating a certain neighbourhood into the complement of the presentation, and we say that we "forbid" the neighbourhood. For some basic $\bar{a} \to \bar{b}$ within this neighbourhood, we must have $\bar{a} \not\to \bar{b}$ in M_e, as witnessed by some first-order atomic ϕ (unless M_e is not total). Then, for some $\bar{c}$, it should be the case that $\phi(\bar{c})$ or $\neg\phi(\bar{c})$, and thus necessarily either $\bar{a} \not\to \bar{c}$ or $\bar{c} \not\to \bar{b}$. Until this happens the construction will proceed in some fixed basic neighbourhood, say $\bar{a} \to \bar{c}$. Once we see ϕ evaluated on $\bar{c}$ (if ever) and says that $\bar{c} \not\to \bar{b}$, then we switch to $\bar{c} \to \bar{b}$ and forbid $\bar{a} \to \bar{c}$. The key here is that we don't have to instantly forbid the neighbourhoods, but M_e must (unless it is not total). We can put sub-neighbourhoods of a given neighbourhood into our effectively open set one-by-one. Thus, we can delay our decision and do the opposite in the group presentation.

The trickier part is nesting the strategies on top of each other. For that, our construction will proceed only within nested clopen subsets extending $\bar{x} \leftrightarrow \bar{y}$, where $\bar{x}$ is an initial segment of ω, $\bar{y}$ is a permutation of $\bar{x}$, and the order of this permutation is 2. If we make sure $\bar{x}\bar{a} \not\to \bar{x}\bar{b}$ in M_e, we can still fix a tuple $\bar{y}\bar{c}$ and repeat the basic diagonalisation idea, as above. It must be the case that either $\bar{x}\bar{a} \not\to \bar{y}\bar{c}$ or $\bar{y}\bar{c} \not\to \bar{x}\bar{b}$ is witnessed by some first-order ϕ, but both events can be restricted to the neighbourhood $\bar{x} \leftrightarrow \bar{y}$. The key here is to choose numbers in $\bar{c}$ to be very large, so that both neighbourhoods $\bar{x}\bar{a} \to \bar{y}\bar{c}$ or $\bar{y}\bar{c} \to \bar{x}\bar{b}$ contain sub-neighbourhoods isolated by finite permutations of order 2. Then the construction can proceed in one of the two neighbourhoods. With some care we will end up with a copy of $\mathbb{Z}_2$. The rest is handled by priority nonsense.

Proof. Fix a computable listing $(M_e)_{e\in\omega}$ of all (partial) computable structures upon the domain ω. We construct a Π^0_1-subgroup P of the standard copy of S_∞, and meet the requirements:

$$P \neq \mathrm{Aut}(M_e),$$

for each e. We will also (globally) ensure that $P \cong \mathbb{Z}_2$.

We will identify finite injective partial maps and the respective basic neighbourhoods in S_∞ determined by their possible extensions.

Definition 3.1. We say that an injective finite map $\bar{x} \to \bar{y}$ is *nice* if it is a finite permutation of an initial segment of ω and has order 2 (i.e., is an involution). We write $\bar{x} \leftrightarrow \bar{y}$ to emphasise that the map and its respective basic neighbourhood are nice.

All our diagonalisation strategies will be working within $(0, 1) \leftrightarrow (1, 0)$. Some of the basic neighbourhoods will be enumerated into the complement of P. If we enumerate a certain neighbourhood into $S_\infty \setminus P$, we say that we *forbid* the neighbourhood. There will be no interaction between the process of approximating Id_ω and the procedure of approximating the only non-identity element of P.

The basic strategy. We describe the basic diagonalisation strategy, for M_e, in isolation. The strategy will be working within a nice $\sigma_e = \bar{x} \leftrightarrow \bar{y}$.

(1) Forbid $\bar{x}n \to \bar{x}(n+1)$, where $n = lth(\bar{x})$. (Note that $\bar{x}n \to \bar{x}(n+1)$ will be forbidden by the construction anyway because of its proximity to Id_ω, so we could simply wait until this happens.)
(2) Wait for M_e to separate some $\bar{x}\bar{a}$ and $\bar{x}\bar{b}$ extending $\bar{x}n$ and $\bar{x}(n+1)$ (and having the same length) by a first-order atomic formula ϕ. Until this happens, if ever, let the construction proceed within the nice neighbourhood $\bar{x}n \leftrightarrow \bar{y}n$. One-by-one, start forbidding all other sub-neighbourhoods $\sigma_e = \bar{x} \leftrightarrow \bar{y}$ of the form $\bar{x}n \to \bar{y}k$, $k \neq n$. (If M_e is total but never gives such a ϕ, then run a back-and-forth argument on extensions of $\bar{x}n$ and $\bar{x}(n+1)$ to build an automorphism of M_e extending $\bar{x}n \to \bar{x}(n+1)$.)
(3) If such a ϕ is found, choose $\bar{c}$ consisting of very large and fresh numbers (and of the same length as $\bar{a}$ and $\bar{b}$). Proceed as follows:
    (a) Extend the finite partial maps $\bar{x}\bar{a} \to \bar{y}\bar{c}$ and $\bar{y}\bar{c} \to \bar{x}\bar{b}$ to [finite] permutations of order 2, let N_1 and N_2 be the respective nice sub-neighbourhoods. Since $\bar{x} \leftrightarrow \bar{y}$ is nice, both N_1 and N_2 belong to $\bar{x} \leftrightarrow \bar{y}$. By the choice of $\bar{c}$, these permutations

have not been forbidden yet. Stop the process of forbidding sub-neighbourhoods of $\bar{x} \leftrightarrow \bar{y}$ initiated at substep (2).

(b) Forbid what is left of $\bar{x}n \leftrightarrow \bar{y}n$. (Note that weaker priority strategies have been working in this neighbourhood.)

(c) Start forbidding extensions of N_2, one-by-one, and let the construction [i.e., all the weaker priority strategies] proceed within N_1. Forbid all basic open neighbourhoods that do not contain Id_ω and are disjoint form N_1 and N_2.

(4) Wait for M_e to evaluate ϕ on $\bar{y}\bar{c}$, thus witnessing either $\bar{x}\bar{a} \not\to \bar{y}\bar{c}$ or $\bar{y}\bar{c} \not\to \bar{x}\bar{b}$ in $\mathrm{Aut}(M_e)$. If this never happens, the construction will forever stay inside N_1.

Case 1: $\bar{x}\bar{a} \not\to \bar{y}\bar{c}$ in $\mathrm{Aut}(M_e)$, as witnessed by ϕ. In this case $\mathrm{Aut}(M_e) \cap N_1 = \emptyset$. Proceed as above to (eventually) completely forbid N_2 and keep all weaker priority actions restricted to N_1. We will see that in this case the only non-zero element of P is inside N_1.

Case 2: $\bar{y}\bar{c} \not\to \bar{x}\bar{b}$ in $\mathrm{Aut}(M_e)$, as witnessed by ϕ. In this case stop forbidding N_2 and forbid what is left of N_1. Choose a nice τ within N_2 and restrict the actions of all weaker priority strategies to τ. Similarly to Case 1, the only non-zero element of P will be contained in τ.

Priority and initialisation. We order the strategies according to the index of the partial computable structure that they are guessing, with smaller indices corresponding to stronger priority. Every time a basic strategy changes its mind about the neighbourhood in which the construction [i.e., the weaker priority strategies]should proceed, we initialise all weaker priority strategies. This is done by picking a new nice neighbourhood σ_i within the current neighbourhood of the higher priority strategy in which it allows the construction to proceed. We also make sure that the diameter of the nice neighbourhood σ_i of the ith strategy is at most 2^{-i} (equivalently, we could require that the domain of the finite nice map contains at least i elements).

Construction. At the beginning of the construction, we will fix a nice basic neighbourhood of Id (say, $(0,1) \leftrightarrow (0,1)$) and some other nice neighbourhood (say, $(0,1) \leftrightarrow (1,0)$) disjoint from it. From this point on, we keep forbidding all (not necessarily nice) sub-neighbourhoods of $(0,1) \leftrightarrow (0,1)$ that do not contain Id_ω. We set $\sigma_0 = (0,1) \leftrightarrow (1,0)$.

Verification. We verify some of the key properties of the construction, stage-by-stage.

Claim 3.2. Suppose M_e is total, and $\bar{x}n \not\to \bar{x}(n+1)$ $\mathrm{Aut}(M_e)$. Then at stage (2) we can find $\bar{x}\bar{a}$ and $\bar{x}\bar{b}$ extending $\bar{x}n$ and $\bar{x}(n+1)$, respectively, and a first-order atomic formula ϕ that separates $\bar{x}\bar{a}$ and $\bar{x}\bar{b}$.

Proof of Claim. Suppose such $\bar{x}\bar{a}$ and $\bar{x}\bar{b}$ and an atomic ϕ do not exist. This means that $\bar{x}n \to \bar{x}(n+1)$ can be extended to an automorphism of M_e in a back-and-forth fashion, contradicting $\bar{x}n \not\to \bar{x}(n+1)$. □

We follow the notation and the terminology used in the construction.

Claim 3.3. Suppose substage (3) is reached. Then there exists a tuple $\bar{c}$ and nice neighbourhoods N_1 and N_2 with the desired properties.

Proof of Claim. Recall that only finitely many basic neighbourhoods can be forbidden at every stage of the construction. In particular, only finitely many sub-neighbourhoods of $\sigma_e = \bar{x} \leftrightarrow \bar{y}$

of the form $\bar{x}n \to \bar{y}k$, $k \neq n$, have been enumerated into the complement of the effectively closed set that we build. In particular, we can choose $\bar{c}$ so that $\bar{x}\bar{a} \to \bar{y}\bar{c}$ has not been forbidden yet. Furthermore, choosing $\bar{c}$ large enough we can ensure that both $\bar{x}\bar{a} \to \bar{y}\bar{c}$ and $\bar{y}\bar{c} \to \bar{x}\bar{b}$ can be extended to finite permutations of order 2 which have not yet been forbidden. This is done by simply setting $\sigma(j) = i$ if $\sigma(i) = j$ already, and by declaring $\sigma(k) = k$ for all other k. □

The importance of choosing $\bar{c}$ very large in (3) is best illustrated by the simple example below.

Example 3.4. In the notation as above, suppose $\bar{x}\bar{a} \to \bar{x}\bar{b}$ is $(0,1,2,7,11) \to (0,1,3,2,5)$. It extends $\bar{x}n \to \bar{y}(n+1)$ which is $(0,1,2) \to (0,1,3)$. Fix A, B, C very large, they could be equal to $100, 101, 102$. Consider $(0,1,2,7,11) \to (1,0,101,102,103)$ and $(1,0,101,102,103) \to (0,1,3,2,5)$. We could extend them to [finite] permutations, say to $(0,1,2,7,11,101,102,103) \to (1,0,101,102,103,2,7,11)$ and $(1,0,2,3,5,101,102,103) \to (0,1,102,101,103,3,2,5)$, respectively. Recall we were slowly forbidding all neighbourhoods in $\bar{x} \leftrightarrow \bar{y}$ except for extensions of $\bar{x}n \leftrightarrow \bar{y}n$, which is $(0,1,2) \to (1,0,2)$ in this particular case. But 101 is large enough so that $(0,1,2,7,11) \to (1,0,101,102,103)$ has not been forbidden yet. The neighbourhood $(1,0,2,3,5,101,102,103) \to (0,1,102,101,103,3,2,5)$ has not been forbidden since 102 is large enough. We could easily extend each of these finite maps to permutations of $\omega \upharpoonright 103$ of order 2 by making all the rest $i < 103$ stable under the permutation. This will give us nice extensions of $(0,1,2,7,11) \to (1,0,101,102,103)$ and $(1,0,101,102,103) \to (0,1,3,2,5)$ which have not been forbidden yet in the construction. (Note that 2 was accidentally mentioned in the domain of the second permutation, due to the choice of $\bar{b}$ and $\bar{a}$. If it was not the case, we'd have to choose a large fresh D and map $2 \leftrightarrow D$, just to make sure the extension is not forbidden.) Note that both neighbourhoods are contained within the basic neighbourhood of $(0,1) \leftrightarrow (1,0)$.

Claim 3.5. Suppose the eth strategy is never initialised after stage s. Regardless of the outcome, there exists an $s' \geq s$ and a nice neighbourhood N^e such that all the weaker priority strategies ($j > e$) perform their actions within N^e.

Proof of Claim. The strategy may never find a ϕ and a pair of witnesses at substage (2), in which case all weaker priority strategies will work within $\bar{x}n \leftrightarrow \bar{y}n$. Otherwise, depending on the outcome, it may either stay within N_1 forever, or it may eventually switch to N_2 and never change the neighbourhood again. □

The basic module of the eth strategy makes sure that no element in the eventually stable neighbourhood N^e can be in $\mathrm{Aut}(M_e)$ if M_e is total. Note that, whenever a strategy is initialised it can pick a nice neighbourhood within the part of S_∞ that has not been forbidden yet by the higher priority strategies. A straightforward inductive argument shows that for every e, the eth strategy eventually never changes its neighbourhood that it keeps unforbidden, and therefore every strategy is eventually never initialised.

The eth strategy ensures that some nice τ_e determined by its stable N^e is the approximation of $P \setminus \{Id_\omega\}$. It follows from the construction that all elements of S_∞ in $(0,1) \leftrightarrow (0,1)$ that do not extend τ_e will be eventually forbidden by the e'th strategy. Also, all neighbourhoods outside $(0,1) \leftrightarrow (0,1)$ that do not contain Id_ω will be forbidden in the construction.

Note that the diameter of the nice eventually stable neighbourhood N_e is at most 2^{-e}, and $N^{e+1} \subset N^e$ for every e. It follows that the intersection of all these eventually stable N^e is a singleton whose only element is the limit of the Δ^0_2 sequence $(\tau_e)_{e\in\omega}$. The singleton describes the only non-Id element Θ of the Π^0_1 set P that we end up with. Note that $\tau_e^2 = Id_{supp(\tau_e)}$, for each e. It follows that $\Theta^2 = Id_\omega$. Thus, $P \cong \mathbb{Z}_2$. □

4. Discrete Π^0_1-presented groups and oligomorphic groups

We know that $\mathbb{Z}_2$ has a complicated Π^0_1 presentation, but it is also clear that $\mathbb{Z}_2$ has a "nice" presentation equal to $\mathrm{Aut}(M)$ for some computable M. This elementary observation is a special case of the more general result: Every discrete Π^0_1-presented group P is isomorphic to $\mathrm{Aut}(M)$ for some computable structure M (Theorem 0.2(2)).

To prove the theorem we analyse the complexity of the orbit equivalence relation. Let G be a subgroup of S_∞. For all $n < \omega$, the group G acts on the collection ω^n of n-tuples of natural numbers; the resulting orbit equivalence relation $\sim_G$ is defined on $\omega^{<\omega}$ by letting $\bar{a} \sim_G \bar{b}$ if $|\bar{a}| = |\bar{b}|$ and there is some $\sigma \in G$ such that $\sigma(\bar{a}) = \bar{b}$. We prove the following:

Proposition 4.1. If G is a discrete, effectively closed subgroup of S_∞, then $\sim_G$ is hyperarithmetic.

(This means that for each n, the orbit equivalence relation for n-tuples is hyperarithmetic, uniformly in n.)

Proposition 4.2. If G is an effectively closed subgroup of S_∞ and $\sim_G$ is hyperarithmetic, then there is a computable structure M such that $G \cong \mathrm{Aut}(M)$.

Theorem 0.2(2) then follows. Proposition 4.2 is itself the conjunction of two lemmas.

Lemma 4.3. Every closed subgroup G of S_∞ is equal to $\mathrm{Aut}(M)$ for some structure M computable from $\sim_G$.

Lemma 4.4. For every hyperarithmetic structure M there is a computable structure N such that $\mathrm{Aut}(N) \cong \mathrm{Aut}(M)$.

Lemma 4.4 is a generalisation of Proposition 2.3. Say that M is Δ^0_γ for some computable ordinal γ. We use a generalisation of Fact 2.4 that allows us to code membership in a Σ^0_α class into an isomorphism type of one of two rigid linear orderings which satisfy the same computable Π_γ infinitary formulas but are separated thereafter. For example, we can produce a copy of $\omega^{\gamma+1}$ if a Σ^0_γ fact holds, and a copy of $\omega^{\gamma+2}$ otherwise; for our purposes, the complexity does not need to be tight. See [GK02, Prop.4.12]. The argument then is the same as that of Proposition 2.3.

Lemma 4.3 is an observation that the standard construction of a structure M such that $G = \mathrm{Aut}(M)$ does in fact give us a structure computable from $\sim_G$, see [Gao09]. For any n and for every $\sim_G$-equivalence class of n-tuples we define an n-ary relation which defines that class. Closure of G is used to show that $\mathrm{Aut}(M) \subseteq G$.

It remains to prove Proposition 4.1.

Proof of Proposition 4.1. Σ^1_1 subsets of ω^ω have the perfect set property in a strongly effective way: if a Σ^1_1 set A does not have a perfect subset then all elements of A are hyperarithmetic, and so by Spector's Σ^1_1 bounding, they are all computable from some $0^{(\gamma)}$ for some fixed computable ordinal γ. (See [Sac90], Thm. 6.2.III.)

By Proposition 1.3, G is a Π^0_2 subset of ω^ω. If all elements of G are $0^{(\gamma)}$-computable, then $0^{(\gamma+2)}$ computes a listing of the elements of G; it then follows that $\sim_G$ is $0^{(\gamma+3)}$-computable. □

This proves Theorem 0.2(2).

Remark 4.5. Note that if G is an effectively closed discrete group, then it has a hyperarithmetical presentation in the sense of computable structure theory (upon the domain of ω). It is easy to see that this observation gives a characterization of discrete effectively closed groups in terms of hyperarithmetically presented groups; we outline the proof of the less obvious implication.

Let G be a countable (discrete) group. Use Cayley's theorem and map $g \in G$ to the permutation $h \mapsto gh$, call it σ_g. The image is discrete: σ_g is isolated by the neighbourhood $e \mapsto g$. This way we obtain an $H \leqq S_\infty$

(topologically) isomorphic to G which is furthermore arithmetical in the diagram of G, in particular $\sim_H$ is HYP. Lemma 4.3 gives a HYP M such that $H \cong Aut(M)$, and Lemma 4.4 allows to build a computable N such that $G \cong H \cong Aut(M) \cong Aut(N)$.

4.1. **Proof of Theorem 0.4.** Recall that a (closed) oligomorphic group is a (closed) subgroup of S_∞ such that for every n there are only finitely many G-orbit equivalence classes of n-tuples. We construct of Σ^1_1, closed oligomorphic subgroup of S_∞ for which $\sim_G$ is not hyperarithmetic.

We work in the admissible structure $L_{\omega_1^{ck}}$. The idea is to use a cofinal ω-sequence $\langle \alpha_n \rangle$ in ω_1^{ck} which is approximated effectively (in the sense of $L_{\omega_1^{ck}}$-computability) with only finitely many changes to each value. We define the orbit equivalence relation $\sim_G$ rather than G. We break up classes of n-tuples into two whenever we see a change in the value of α_n; this happens only finitely many times, so at the end we get only finitely many classes of n-tuples. The entire equivalence relation is not hyperarithmetic, as we can recover the sequence $\langle \alpha_n \rangle$ from it. To ensure that the equivalence relation is indeed the orbit equivalence relation of a closed group we need to ensure that it is invariant under permutations, taking subsequences, and has the back-and-forth property. This can be done dynamically, but in fact it is easy to give an explicit definition of the relation.

To the details. For a non-decreasing sequence $\mathbf{k} = \langle k_n \rangle$ of natural numbers define an equivalence relation $\sim_{\mathbf{k}}$ on $\omega^{<\omega}$. For $\bar{a}, \bar{b} \in \omega^{<\omega}$ of the same length n, if they are both injective, then we let $\bar{a} \sim_{\mathbf{k}} \bar{b}$ if for every $m < n$ there are at least $n - m$ many $i < n$ such that $a_i = b_i \mod 2^{k_m}$. We then extend this to non-injective tuples in the obvious way.

We first observe:

- $\sim_{\mathbf{k}}$ is an equivalence relation.
- If $\bar{a} \sim_{\mathbf{k}} \bar{b}$ then for any subsequence $\bar{a}'$ of $\bar{a}$ and $\bar{b}'$ of $\bar{b}$ chosen in the same way, $\bar{a}' \sim_{\mathbf{k}} \bar{b}'$.
- $\sim_{\mathbf{k}}$ is invariant under permutations: for any n-tuples $\bar{a}, \bar{b}$ and permutation σ of n, if $\bar{a} \sim_{\mathbf{k}} \bar{b}$ then $\bar{a} \circ \sigma \sim_{\mathbf{k}} \bar{b} \circ \sigma$.
- $\sim_{\mathbf{k}}$ has the back and forth property: if $\bar{a} \sim_{\mathbf{k}} \bar{b}$ then for all $c < \omega$ there is some $d < \omega$ such that $\bar{a}c \sim_{\mathbf{k}} \bar{b}d$.
- For every n there are only finitely many $\sim_{\mathbf{k}}$-equivalence classes of n-tuples.

We then let $G_{\mathbf{k}}$ be the collection of all $f \in S_\infty$ such that for all $\bar{a}$ and $\bar{b}$, if $f(\bar{a}) = \bar{b}$ then $\bar{a} \sim_{\mathbf{k}} \bar{b}$. It is a closed subgroup of S_∞; the properties just listed ensure that $\sim_{\mathbf{k}}$ is the orbit equivalence relation of the action of $G_{\mathbf{k}}$, and that this action is oligomorphic.

We now start to work effectively. Since the Σ_1 projectum of $L_{\omega_1^{ck}}$ is ω, there is a $\Delta_2(L_{\omega_1^{ck}})$ increasing sequence $\langle \alpha_n \rangle_{n<\omega}$ which is cofinal in ω_1^{ck}; see [Sac90] for an excellent exposition of higher recursion theory. In fact, $\langle \alpha_n \rangle$ has a *finite-change approximation* (see [BGM17]): there is a $\Delta_1(L_{\omega_1^{ck}})$ array (that is, a $L_{\omega_1^{ck}}$-computable array) $\langle \alpha_{n,s} \rangle_{n<\omega, s<\omega_1^{ck}}$ such that writing $\alpha_{n,\omega_1^{ck}}$ for α_n, we have:

- For all limit ordinals $s \leq \omega_1^{ck}$, for all n, $\alpha_{n,s} = \lim_{t \to s} \alpha_{n,t}$; and
- For every $n < \omega$, there are only finitely many stages $s < \omega_1^{ck}$ such that $\alpha_{n,s} \neq \alpha_{n,s+1}$.

Given such a sequence we define, for each $s \leq \omega_1^{ck}$,

$$k_{n,s} = \sum_{m \leq n} \#\{t < s \,:\, \alpha_{m,t} \neq \alpha_{m,t+1}\}.$$

this defines sequences $\mathbf{k}_s$ for $s \leq \omega_1^{ck}$. Our final equivalence relation is $\sim_{\mathbf{k}} = \sim_{\mathbf{k}_{\omega_1^{ck}}}$. The fact that we used powers of 2 means that as for $t < s$, for all n, $k_{n,t} \leq k_{n,s}$, the equivalence relation

$\sim_{\mathbf{k}_{n,t}}$ refines the equivalence relation $\sim_{\mathbf{k}_{n,s}}$. This shows that $\sim_{\mathbf{k}}$ is $\Pi_1(L_{\omega_1^{ck}})$, that is, it is Σ^1_1. It follows that $G_{\mathbf{k}}$ is Σ^1_1 as well.

The final relation $\sim_{\mathbf{k}}=\sim_{\mathbf{k}_{\omega_1^{ck}}}$ is not hyperarithmetic: we can see that $\omega_1^{\sim_{\mathbf{k}}} > \omega_1^{ck}$, as in an effective fashion over $L_{\omega_1^{\sim_{\mathbf{k}}}}[\sim_{\mathbf{k}}]$ we can recover $\langle \alpha_n \rangle$: for all $n < \omega$ and $s < \omega_1^{ck}$, if the number of $\sim_{\mathbf{k}_s}$-equivalence classes of n-tuples is the same as that of $\sim_{\mathbf{k}}$, then $\alpha_n = \alpha_{n,s}$.

References

[AK00] C. 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.

[BC08] Paul Brodhead and Douglas Cenzer. Effectively closed sets and enumerations. *Arch. Math. Logic*, 46(7-8):565–582, 2008.

[BGM17] Laurent Bienvenu, Noam Greenberg, and Benoit Monin. Continuous higher randomness. *J. Math. Log.*, 17(1):1750004, 53, 2017.

[BHW08] V. Brattka, P. Hertling, and K. Weihrauch. A tutorial on computable analysis. In *New computational paradigms*, pages 425–491. Springer, New York, 2008.

[EG00] Y. Ershov and S. Goncharov. *Constructive models*. Siberian School of Algebra and Logic. Consultants Bureau, New York, 2000.

[Gao09] Su Gao. *Invariant descriptive set theory*, volume 293 of *Pure and Applied Mathematics (Boca Raton)*. CRC Press, Boca Raton, FL, 2009.

[GK02] S. Goncharov and J. Knight. Computable structure and antistructure theorems. *Algebra Logika*, 41(6):639–681, 757, 2002.

[GMKT] Noam Greenberg, Alexander G. Melnikov, Julia F. Knight, and Daniel Turetsky. Uniform procedures in uncountable structures. *To appear.*

[GR93] Xiaolin Ge and J. Ian Richards. Computability in unitary representations of compact groups. In *Logical methods (Ithaca, NY, 1992)*, volume 12 of *Progr. Comput. Sci. Appl. Logic*, pages 386–421. Birkhäuser Boston, Boston, MA, 1993.

[HK14] Kojiro Higuchi and Takayuki Kihara. On effectively closed sets of effective strong measure zero. *Ann. Pure Appl. Logic*, 165(9):1445–1469, 2014.

[Jar74] Moshe Jarden. Algebraic extensions of finite corank of Hilbertian fields. *Israel J. Math.*, 18:279–307, 1974.

[LR81] Peter La Roche. Effective Galois theory. *J. Symbolic Logic*, 46(2):385–392, 1981.

[McN15] Timothy H. McNicholl. A note on the computable categoricity of ℓ^p spaces. In *Evolving computability*, volume 9136 of *Lecture Notes in Comput. Sci.*, pages 268–275. Springer, Cham, 2015.

[Mel] A. Melnikov. Computable topological groups and Pontryagin duality. To appear in Transactions of The American Mathematical Society.

[Mel13] Alexander G. Melnikov. Computably isometric spaces. *J. Symbolic Logic*, 78(4):1055–1085, 2013.

[MM] Alexander G. Melnikov and Antonio Montalbán. Computable polish group actions. *To appear.*

[MN79] G. Metakides and A. Nerode. Effective content of field theory. *Ann. Math. Logic*, 17(3):289–320, 1979.

[MN13] Alexander G. Melnikov and André Nies. The classification problem for compact computable metric spaces. In *The nature of computation*, volume 7921 of *Lecture Notes in Comput. Sci.*, pages 320–328. Springer, Heidelberg, 2013.

[MN16] Alexander G. Melnikov and Keng Meng Ng. Computable structures and operations on the space of continuous functions. *Fund. Math.*, 233(2):101–141, 2016.

[MS] Timothy H. McNicholl and D. M. Stull. The isometry degree of a computable copy of l^p. *To appear.*

[PER89] Marian B. Pour-El and J. Ian Richards. *Computability in analysis and physics*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1989.

[Pon66] L. S. Pontryagin. *Topological groups*. Translated from the second Russian edition by Arlen Brown. Gordon and Breach Science Publishers, Inc., New York-London-Paris, 1966.

[Rei08] Jan Reimann. Effectively closed sets of measures and randomness. *Ann. Pure Appl. Logic*, 156(1):170–182, 2008.

[Sac90] Gerald E. Sacks. *Higher recursion theory*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1990.

[Smi79] Rick L. Smith. *THE THEORY OF PROFINITE GROUPS WITH EFFECTIVE PRESENTATIONS*. ProQuest LLC, Ann Arbor, MI, 1979. Thesis (Ph.D.)–The Pennsylvania State University.

[Smi81] Rick L. Smith. Effective aspects of profinite groups. *J. Symbolic Logic*, 46(4):851–863, 1981.