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 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 closed 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.

1. Introduction

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. 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; an example of study in the more general direction is [GR93]. However, the general theory of computable Polish groups remains at an initial stage. Recently there has been an increased 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

The first author and third authors were 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.

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.

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 computable. Our main goal is to investigate automorphism groups of computable algebraic structures. For this purpose we consider *effectively closed* subgroups of S_∞. These are the subgroups of S_∞ which are effetively closed subsets of a natural computable metric space structure on S_∞, which we define in Section 2 below. Equivalently (Corollary 2.4) they are the intersection with S_∞ of effectively closed subsets of Baire space ω^ω. Since the domain of a computable structure $\mathcal{M}$ is ω, its automorphism group $\operatorname{Aut}(\mathcal{M})$ is a subgroup of S_∞, and in fact it is effectively closed.

It is well-known that every closed subgroup of S_∞ is equal to the automorphism group of some countable algebraic structure on ω (see for example [Gao09, Thm.2.4.4]). It is natural to ask:

> Is every effectively closed group equal to $\operatorname{Aut}(\mathcal{C})$ 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 $\operatorname{Aut}(\mathcal{C})$ for some computable structure $\mathcal{C}$. An *effectively closed copy* of a topological group G is an effectively closed subgroup of S_∞ (topologically) isomorphic to G.

Theorem 1.1.

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

The main difficulty in the proof of Theorem 1.1(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 1.1(2) can be extended to general effectively closed groups. The proof of Theorem 1.1(2) is related to a characterisation of discrete, effectively closed subgroups of S_∞ (Remark 5.5), and the complexity of their orbit equivalence relations.

The concept of effectively closed subgroups of S_∞ is natural in its own right; closed subgroups of topological groups, including S_∞, are studied in depth; and 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 will compare this concept with the notions of a *computable Polish* *group* [MM, Mel] and a *recursively presented profinite group* [LR81, Smi81] which we mentioned above (we give the formal definitions in Section 2). Every recursively presented profinite group is computable Polish, but there are computable Polish profinite groups with no recursive presentation [Mel].

Theorem 1.2.

(1) There is an effectively closed, profinite subgroup of S_∞ that has no computable Polish copy (and therefore, no recursive presentation).
(2) Every profinite computable Polish group has an effectively closed copy.

In particular, Theorem 1.2(1) shows that the notion of effectively closed subgroups of S_∞ is new, while Theorem 1.2(2) establishes a connection between this notion and computable Polish groups. The proof of Theorem 1.2(2) in fact gives a group $\mathrm{Aut}(\mathcal{M})$ for a computable structure $\mathcal{M}$, which in light of Theorem 1.1(1) is stronger.

We end the paper by looking at the class of oligomorphic groups. These are the closed subgroups G of S_∞ for which for every n there are only finitely many G-orbit equivalence classes of n-tuples (every subgroup of S_∞ acts on ω^n in the natural way). 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 the orbit equivalence relation $\sim_G$ will be relatively simple: $\mathbf{0}^{(\omega)}$-computable; if the signature of M is finite and M has quantifier elimintation, then $\sim_G$ is computable (uniformly in n). We approach the question "how complicated can $\sim_G$ be for an effectively closed oligomorphic G?" If we could construct an effectively closed oligomorphic group with complicated $\sim_G$, we would get another example for our main result Theorem 1.1(1). Oligomorphic groups lie at the other extreme from profinite groups, for which every orbit equivalence class is finite. Our initial hope was that it might perhaps be easier to handle them. This hope was ill-founded, yet the effective content of oligomorphic groups turned out to be interesting in its own right.

The natural upper bound for the complexity of $\sim_G$ for effectively closed groups G is Σ^1_1; this upper bound does not change even if the group G is Σ^1_1. Even in that case, if G is oligomorphic, then for each n, the restriction of $\sim_G$ to n-tuples must be hyperarithmetic. Interestingly enough, this fact lacks uniformity in the following sense.

Theorem 1.3. 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 an effectively closed oligomorphic group can witness Theorem 1.1(1).

The structure of this paper is as follows. The short Section 2 contains formal definitions, a description of the natural computable Polish presentation of S_∞, and the equivalent definition of effectively closed subgroups of S_∞ using the inherited topology from Baire space. We arrange proofs according to the methods used. Section 3 contains the proof of Theorem 1.2, Section 4 the proof of Theorem 1.1(1), and Section 5 the proofs of Theorem 1.1(2) and Theorem 1.3.

2. 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 2.1. A *computable Polish (metric) space* is a triple $(M, d, (\alpha_i)_{i\in\omega})$, where M is a Polish space, d is a complete compatible metric on M, 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*. Baire space, ω^ω, under the usual ultrametric forms a computable Polish space. The basic open balls are the usual clopen neighbourhoods: the ones determined by finite sequences of natural numbers.

A subset U of a computable Polish space is *effectively open* if it is the uniform union of basic open sets. That is, if there is a c.e. set $W \subseteq \omega \times \mathbb{Q}^+$ such that $U = \bigcup_{(i,q)\in W} B(\alpha_i, q)$. Note that we are not required (and outside the compact case, not always able) to enumerate all the basic open balls which are subsets of U. A set is *effectively closed* if its complement is effectively open.

If $\mathbb{X}$ and $\mathbb{Y}$ are computable Polish spaces, then we say that a map $F\colon \mathbb{X} \to \mathbb{Y}$ is *computable* if for every effectively open set $U \subseteq \mathbb{Y}$, $F^{-1}[U]$ is an effectively open subset of $\mathbb{X}$, *uniformly*: from an index for U we effectively obtain an index for $F^{-1}[U]$. It suffices to enumerate $F^{-1}[B]$ uniformly for the basic open balls B.

For the following definition, observe that the product of two (or more) computable Polish spaces is itself computable Polish.

Definition 2.2 ([MM])**.** A *computable Polish group* is a computable Polish space which is a topological group for which the group operation $\cdot$ and inverse $^{-1}$ are computable.

We now discuss the infinite permutation group $S_\infty \subset \omega^\omega$. The usual ultrametric d inherited from Baire space is left-invariant for S_∞ but is not complete: S_∞ is not a closed subset of ω^ω. A complete metric which gives the same topology is as follows: for $f, g \in S_\infty$ let

$$D(f, g) = \frac{d(f, g) + d(f^{-1}, g^{-1})}{2}.$$

We let the special points be the permutations of ω with finite support. This makes S_∞ a computable Polish group.

The compatibility of the metrics d and D is effective. Formally, what this means is that the identity map from (S_∞, D) to (ω^ω, d) is computable, and its inverse is a partial computable map. We work more concretely by giving finite descriptions to effectively open sets, including all the basic open balls. For any finite partial map σ from ω to ω, $[\sigma]$ is defined to be the set of $f \in \omega^\omega$ which agree with σ: $f(n) = \sigma(n)$ for all $n \in \operatorname{dom}\sigma$. For each finite partial map σ, $[\sigma]$ is an effectively open subset of ω^ω; every basic open set is of the form $[\sigma]$ for some σ (indeed, precisely for those finite maps whose domains are finite initial segments of ω); and a set $U \subseteq \omega^\omega$ is effectively open if and only if there is a c.e. set W of finite partial functions such that $U = [W] = \bigcup\{[\sigma] : \sigma \in W\}$.

Proposition 2.3. A set $V \subseteq S_\infty$ is effectively open if and only if there is an effectively open set $U \subseteq \omega^\omega$ such that $V = U \cap S_\infty$.

Proof. Indeed the proof will show that the translation is uniform. A basic open ball in S_∞ is determined by specifying the values of a permutation on a finite initial segment of ω, and the values of its inverse on a finite initial segment of ω. That is, a basic open ball is of the form

$$[\sigma; \tau] = \left\{ f \in S_\infty \,:\, \sigma \prec f \;\&\; \tau \prec f^{-1} \right\}$$

where $\sigma, \tau \in \omega^{<\omega}$, that is, are finite partial functions whose domain is an initial segment of ω. Of course this is empty unless both σ and τ are injective, σ agrees with τ^{-1}, and τ agrees with σ^{-1}. Now given a finite map ρ from ω to ω,

$$[\rho] \cap S_\infty = \bigcup \left\{ [\sigma; \tau] \,:\, \sigma \text{ extends } \rho \right\},$$

whereas

$$[\sigma; \tau] = [\sigma \cup \tau^{-1}] \cap S_\infty. \qquad \square$$

Corollary 2.4. A set $P \subseteq S_\infty$ is effectively closed if and only if there is a co-c.e. set C of finite partial injective functions such that

$$P = \left\{ f \in S_\infty \,:\, f \text{ disagrees with every } \sigma \in C \right\}.$$

Corollary 2.4 will be our "working definition" for defining effectively closed subgroups of S_∞.

We turn to another notion of effectiveness for profinite groups. Recall that a group is profinite if it is isomorphic to the inverse limit of a system

$$G_0 \leftarrow G_1 \leftarrow G_2 \leftarrow \dots$$

where each G_i is finite group and all the maps are surjective group homomorphisms.

Definition 2.5 ([LR81, Smi81]). A *recursive presentation* of a profinite group P is a uniformly computable inverse system of finite groups, with surjective homomorphisms, whose inverse limit is isomorphic to P.

We note that the use of the term "recursive presentation" is different from the one in combinatorial group theory [LS77] (which applies only in the discrete case).

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].

3. Effectively closed vs. computable Polish

Proof of Theorem 1.2(1). We construct an effectively closed subgroup of S_∞ that has no computable Polish copy. We will be using the result below:

Fact 3.1 ([Mel], Cor. 1.8). Every computable Polish profinite group P has a $0'$-computable presentation.

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 copy then S is Σ^0_2. To see this, by Fact 3.1 and [Mel, Thm.1.9] there is a $0'$-computable copy – 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_S (see the book [Pon66] for more on Pontryagin's duality theory). Using this copy 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. More specifically, fix a recursive predicate R such that $p\in S \iff \exists^\infty x R(p,x)$ for every 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 1.2(2). Let P be a computable Polish profinite group. We need to produce an effectively closed copy of P. By Fact 3.1, there is a $0'$-recursive presentation of P. We will use a fully relativised form of the fact below.

Fact 3.2 (LaRoche [LR81])**.** Every recursively presented profinite group is isomorphic to $\mathrm{Gal}(K/N)$, where K is a computable algebraic extension of $\mathbb{Q}$, and N is a computable subfield of K.[1]

In fact, N is a fixed computable field that corresponds to a natural recursive presentation of the free profinite group upon countably many generators, see [LR81]. That is, N does not depend on K and on the recursive profinite group. This fact can be partially relativised to $0'$-recursive profinite groups. Note that N stays computable under this relativisation. Fix a $0'$-computable field 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

[1]We note here that Corollary after Prop. 2 on p. 390 of [LR81] is stated in terms of co-r.e. presented profinite groups and r.e.-presented fields. But (as expected) recursive profinite groups correspond to recursive fields, see Thm. 1(2) of the same paper.

$\mathrm{Aut}(F) \cong \mathrm{Aut}(K/N)$ (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. We name each element of N by a a singleton unary predicate and replace the field operations by their graphs. Note that $\mathrm{Aut}(F) \cong \mathrm{Aut}(K/N)$. The next step is a version of Marker's existential extension which preserves the automorphism group.

Proposition 3.3. 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 Ash-Knight technique of coding complexity into pairs of structures. What we observe is that the structures involved can be rigid.

Fact 3.4 ([AK90])**.** 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.

Proof Sketch of Fact 3.4. Fix an increasing sequence $x_0, x_1, \dots$ of type ω and a computable predicate P such that $y \in S \iff \exists^\infty z P(y,z)$. For each i, make the interval $[x_i, x_{i+1})$ isomorphic to $\omega \restriction n$ if $\exists^n z P(y,z)$. □

We assume that the underlying set of A is a computable set, with a computable language. Define B as follows. The universe of A will be a subset of the universe of B. For each n, every n-ary relation P of A, and every tuple $\bar{a} \in A^n$ we fix an infinite set $C^P_{\bar{a}}$. These subsets will be pairwise disjoint for different choices of $\bar{a}$ and P and each disjoint from A. This defines the universe of B. The relations of B are:

- a unary predicate picking out the elements of A;
- for each n-ary relation symbol P of A, the $(n+1)$-ary relation $y \in C^P_{\bar{x}}$;
- A binary relation $<$ which linearly orders each $C^P_{\bar{a}}$; if $P^A(\bar{a})$ holds, then $C^P_{\bar{a}} \cong \omega$; if not, then $C^P_{\bar{b}} \cong \omega^2$.

Fact 3.4 implies that B has a computable copy.

We show that $\mathrm{Aut}(A) \cong \mathrm{Aut}(B)$. Let $f \in \mathrm{Aut}(B)$. Then $f \restriction A$ is an automorphism of A: for any relation symbol P of A and $\bar{a} \in A$, f must map $C^P_{\bar{a}}$ onto $C^P_{f(\bar{a})}$, and $P^A(\bar{a})$, $P^A(f(\bar{a}))$ depend on the order-type of $C^P_{\bar{a}}$ and $C^P_{f(\bar{a})}$, which are isomorphic (by f). Further, f is determined by $f \restriction A$: as observed, f must map $C^P_{\bar{a}}$ to $C^P_{f(\bar{a})}$, and as ω and ω^2 are rigid, the order-isomorphism between $C^P_{\bar{a}}$ and $C^P_{f(\bar{a})}$ is unique. Similarly we observe that every automorphism of A can be extended to an automorphism of B.

Let $\Phi\colon \mathrm{Aut}(B) \to \mathrm{Aut}(A)$ be the restriction map, $\Phi(f) = f \restriction A$. This is a group isomorphism; it remains to see that it is topological. In the slightly less immediate direction, we need to check that it is an open map. We use Corollary 2.4. Let σ be a finite injective map from ω to ω, which determines a basic open set $[\sigma] \cap \mathrm{Aut}(B)$. 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^2 are rigid, mapping $\bar{a}$ to $\bar{b}$ is equivalent to mapping q to p. □

This completes the proof of Theorem 1.2(2). □

4. Proof of Theorem 1.1

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 (i.e., there is no automorphism of M_e extending this finite map), as witnessed by some first-order atomic ϕ. Nonetheless, keep in mind the possibility that M_e is not total, in this case we will have to wait forever. For now, assume M_e responds. 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 that ϕ and says that $\bar{c} \not\to \bar{b}$ (if ever), 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 $\bar{x} \leftrightarrow \bar{y}$ (the notation should be self-explanatory; to be clarified in Def. 4.1), 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 identify finite injective partial maps and the respective basic neighbourhoods in S_∞ determined by their possible extensions.

Definition 4.1. 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 there exists an automorphism of M_e extending $\bar{x}n \to \bar{x}(n+1)$. To build such an automorphism, run a back-and-forth argument on extensions of $\bar{x}n$ and $\bar{x}(n+1)$.)
(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 sub-step (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_ω (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 4.2. Suppose M_e is total, and $\bar{x}n \not\to \bar{x}(n+1)$ in $\mathrm{Aut}(M_e)$. 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 4.3. 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 $\bar{c}$ with elements 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 4.4. 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 $\omega \restriction 103$ 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 4.5. 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. $\square$

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, $P \cong \mathbb{Z}_2$. $\square$

5. Discrete effectively closed groups and oligomorphic groups

We know that $\mathbb{Z}_2$ has a complicated effectively closed copy, but it is also clear that $\mathbb{Z}_2$ has a "nice" copy equal to $\mathrm{Aut}(M)$ for some computable structure M. This elementary observation is a special case of the more general result: Every discrete effectively closed group P is isomorphic to $\mathrm{Aut}(M)$ for some computable structure M (Theorem 1.1(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 5.1. 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 5.2. 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 1.1(2) then follows. Proposition 5.2 is itself the conjunction of two lemmas.

Lemma 5.3. Every closed subgroup G of S_∞ is equal to $\mathrm{Aut}(M)$ for some structure M computable from $\sim_G$.

Lemma 5.4. For every hyperarithmetic structure M there is a computable structure N such that $\mathrm{Aut}(N) \cong \mathrm{Aut}(M)$.

Lemma 5.4 is a generalisation of Proposition 3.3. Say that M is Δ^0_γ for some computable ordinal γ. We use a generalisation of Fact 3.4 that allows us to code membership in a Σ^0_γ set 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 [AK90]. The argument then is the same as that of Proposition 3.3.

Lemma 5.3 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 5.1.

Proof of Proposition 5.1. 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 Corollary 2.4, 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 1.1(2).

Remark 5.5. 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 $e \mapsto g$. In this way we obtain a closed subgroup H of S_∞ (topologically) isomorphic to G, which is furthermore arithmetical in the diagram of G, in particular $\sim_H$ is HYP. Lemma 5.3 gives a HYP M such that $H \cong \mathrm{Aut}(M)$, and Lemma 5.4 allows us to build a computable N such that $\mathrm{Aut}(M) \cong \mathrm{Aut}(N)$.

5.1. **Proof of Theorem 1.3.** Recall that a (closed) oligomorphic group is a (closed) subgroup of S_∞ such that for every n there are only finitely many G-orbits of n-tuples. We construct a Σ^1_1, closed oligomorphic subgroup of S_∞ for which $\sim_G$ is not hyperarithmetic.

We will construct an array of uniformly computable relations R^n_k with R^n_k being n-ary. For an injective sequence $\bar{a} = \langle a_1, \ldots, a_n \rangle$ of natural numbers and another sequence $\mathbf{k} = \langle k_1, \ldots, k_n \rangle$ of natural numbers we write $\mathrm{tp}_{\mathbf{k}}(\bar{a})$ for the atomic type of the tuple $\bar{a}$ in the structure consisting of the relations R^m_k for $m \leq n$ and $k < k_m$. Namely, the type is determined by the values of such R^m_k on tuples $(a_{i_1}, a_{i_2}, \ldots, a_{i_m})$ for distinct $i_1, \ldots, i_m \leq n$ (not necessarily in order).

Lemma 5.6. There is a sequence of uniformly computable relations R^n_k on ω such that:

(1) For all finite sequences $\mathbf{k}$, every consistent $\mathbf{k}$-type is realised; and
(2) For all $\mathbf{k}$, injective $\bar{a}$ and $\bar{b}$, if $\mathrm{tp}_{\mathbf{k}}(\bar{a}) = \mathrm{tp}_{\mathbf{k}}(\bar{b})$ then for all l and $c \notin \{a_1, \ldots, a_n\}$ there is some $d \notin \{b_1, \ldots, b_n\}$ such that $\mathrm{tp}_{\mathbf{k}\hat{\ }l}(\bar{a}, c) = \mathrm{tp}_{\mathbf{k}\hat{\ }l}(\bar{b}, d)$.

Proof. We do a recursive construction by stages. At sage s we will have determined finitely many values of (the characteristic functions of) finitely many relations R^n_k. The requirements to meet are:

- For each $\mathbf{k}$, every consistent $\mathbf{k}$-type is realised;
- For each n and k, for each injective n-tuple $\bar{a}$, the value $R^n_k(\bar{a})$ is decided;
- the back-and-forth property (2) above.

We put all requirements in an ω-list. The first kind of requirement is met by choosing fresh values for an n-tuple and declaring relations accordingly. For a requirement of the second kind encountered at a stage s, if $R^n_k(\bar{a})$ is not yet defined by stage s, then we can choose either value. If a requirement of the third kind is encountered at stage s, if $\mathrm{tp}_{\mathbf{k}}(\bar{a})$ or $\mathrm{tp}_{\mathbf{k}}(\bar{b})$ have not yet been completely determined, then we decide them and make them distinct. Otherwise, we meet the requirement by choosing d fresh, a number which has not been encountered yet in the construction; so we are completely free to determine the value of any relation on any tuple involving d, and we do it to match $\mathrm{tp}_{\mathbf{k}\hat{\ }l}(\bar{a}, c)$. □

For an *infinite* sequence $\mathbf{k} = \langle k_1, k_2, \ldots \rangle$ of natural numbers, we let $\mathcal{M}_{\mathbf{k}}$ be the structure consisting of the relations R^n_k for all n and all $k < k_n$. The back-and-forth property implies:

Lemma 5.7. Injective n-tuples $\bar{a}$ and $\bar{b}$ have the same $\mathrm{Aut}(\mathcal{M}_{\mathbf{k}})$-orbit if and only if they have the same $\mathbf{k} \upharpoonright n$-type.

(Here we abuse notation and let $\mathbf{k} \upharpoonright n = \langle k_1, k_2, \ldots, k_n \rangle$.) Also note that for each n, there are only finitely many $\mathbf{k} \upharpoonright n$-types of injective n-tuples, and this implies that the action of $\mathrm{Aut}(\mathcal{M}_{\mathbf{k}})$ is oligomorphic. Further, the sequence $\mathbf{k}$ is computable if we are given the number of $\mathrm{Aut}(\mathcal{M}_{\mathbf{k}})$-orbits of n-tuples for each n. Hence:

Lemma 5.8. For any $\mathbf{k}$, the sequence $\mathbf{k}$ is computable given the jump of the $\mathrm{Aut}(\mathcal{M}_{\mathbf{k}})$-orbit equivalence relation.

The last piece is the following. Call a sequence $\mathbf{k}$ *left*-Π^1_1 if its undergraph $\{(n, k) : k < k_n\}$ is Π^1_1.

Lemma 5.9. There is a left Π^1_1-sequence $\mathbf{k}$ which collapses ω_1^{ck}, that is, $\omega_1^{\mathbf{k}} > \omega_1^{ck}$.

In particular $\mathbf{k}$ is not hyperarithmetic.

Proof. 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 for example [Sac90]. 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 $\alpha_{n,s}\neq\alpha_{n,s+1}$.

We then let $k_n=\#\left\{s<\omega_1^{ck} : \alpha_{n,s+1}\neq\alpha_{n,s}\right\}$ be the mind-change function for this approximation. If we are given $\mathbf{k}=\langle k_n\rangle$ then we can find α_n by running the approximation $\langle\alpha_{n,s}\rangle$ and waiting for the appropriate number of changes. This means that $\langle\alpha_n\rangle$ is $\Delta_1(L_{\omega_1^{\mathbf{k}}})(\mathbf{k})$, and so $\omega_1^{\mathbf{k}}>\omega_1^{ck}$. □

Let $\mathbf{k}$ be left-Π^1_1. Then $\mathrm{Aut}(\mathcal{M}_{\mathbf{k}})$ is closed Σ^1_1: we remove a neighbourhood $\bar{a}\mapsto\bar{b}$ when we see $\mathbf{k}$ increase so that $\mathrm{tp}_{\mathbf{k}\upharpoonright n}(\bar{a})\neq\mathrm{tp}_{\mathbf{k}\upharpoonright n}(\bar{b})$. If $\mathbf{k}$ is not hyperarithmetic, then by Lemma 5.8, neither is the orbit equivalence relation for the action of $\mathrm{Aut}(\mathcal{M}_{\mathbf{k}})$.

References

[AK90] C. J. Ash and J. F. Knight. Pairs of recursive structures. *Ann. Pure Appl. Logic*, 46(3):211–234, 1990.

[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.

[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.

[LS77] R.C. Lyndon and P.E. Schupp. *Combinatorial group theory*. sv, 1977.

[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.

[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.