Computable permutations and word problems

L'Enseignement Mathématique (2) 64 (2018), 143–160 DOI 10.4171/LEM/64-1/2-6

Computable permutations and word problems

Andrey Morozov and Paul Schupp

Abstract. This is an expository paper whose goal is to explain some interesting interconnections between group theory and the theory of computability. Let $\mathcal{C}$ denote the group of all computable permutations of $\mathbb{N}$. A basic question is: What can one say about the finitely generated subgroups of $\mathcal{C}$? Partially answering this question involves ideas from the theory of computability such as Turing degrees and truth-table degrees. We want to make this paper accessible to both group theorists and computability theorists so we carefully explain all the needed concepts.

Mathematics Subject Classification (2010). Primary: 20F36; Secondary: 03D35.

Keywords. Computable permutation, Turing degree, truth-table degree, HNN extension.

1. Introduction

We consider the group $\mathcal{C}$ of all computable permutations of $\mathbb{N}$, so we have to begin with a precise definition of "computable". In his famous 1936 paper Turing [Tur1] formulated the idea of the universal digital computer, the Turing machine. It is not necessary to have in mind the exact details of how Turing machines work and one can think of such a machine as an ordinary computer with a fixed program but with infinite memory and no time limit on computations. The crucial fact is that a Turing machine is completely defined by its finite program and that there are therefore only countably many Turing machines and one can make an effective list of all of them. (For example, see [Soa] for precise details.) One does not need to have mind the exact working details of a Turing machine when thinking about computability and computability theorists don't worry about them when proving theorems. It is now accepted that *computable by a Turing machine* is the correct precise definition of the word "computable".

There are essentially two ways to use Turing machines. First of course, to compute functions. If Σ and Δ are finite alphabets then Σ^* and Δ^* denote the set of all words over Σ and Δ respectively. A function $f\colon \Sigma^* \to \Delta^*$ is

computable if there exists a Turing machine M which, when started with an arbitrary input $w \in \Sigma$ halts with output $f(w)$. A set $S \subseteq \Sigma^*$ is *computable* if its characteristic function is computable.

The second basic way of using Turing machines is as enumerators. A set $S \subseteq \Sigma^*$ is *computably enumerable* if there exists a Turing machine with a special output tape, which successively writes elements of Σ^*, say separated by spaces, on its output tape, and the set of words eventually listed is exactly S. In other words, one can effectively list the elements of S. Note that there is no constraint on the order in which elements are listed and that a given element may be listed many times, even infinitely often.

A first basic lemma of computability theory is that a set $S \subseteq \Sigma^*$ is computable if and only if both S and its complement $\overline{S}$ are computably enumerable. If both S and its complement $\overline{S}$ are computably enumerable, we can start running both enumerations. Any given word w will occur in exactly one of the enumerations and when it does, we can correctly answer if $w \in S$. If S is computable, then an enumeration of either S or $\overline{S}$ is obtained by enumerating Σ^* and listing according to whether or not a word is in S. (Note that this proof uses no technical details about how Turing machines work.)

Returning to the group $\mathcal{C}$, note that $\mathcal{C}$ is countable since there are only countably many Turing machines. The general question which we consider is: What can one say about the finitely generated subgroups of $\mathcal{C}$? The *word problem*, $WP(G)$, of a finitely generated group $G = \langle X; R \rangle$ is the set of all words over the group alphabet $\Sigma = X \cup X^{-1}$ which are equal to the identity element of G in the given presentation. The word problem for G is *computable*, (*solvable*, *decidable*) if $WP(G)$ is a computable set. Although the exact set $WP(G)$ depends on the given presentation, whether or not the word problem is computable, indeed its exact complexity, does not depend on the presentation. Indeed, if $H = \langle Y; S \rangle$ is a finitely generated presentation of a group isomorphic to a subgroup of G then there is an embedding $\phi : H \to G$. So a word $u = z_1 \cdots z_n$ equals the identity in H exactly if $\phi(u) = \phi(z_1) \cdots \phi(z_n)$ equals the identity in G, and $\phi(u)$ is calculable in linear time.

To say that we are *given* finitely many generators of a subgroup G of $\mathcal{C}$, we mean that we are given the algorithms (Turing machines) which compute the permutations. From this information we cannot decide when an element is the identity, but we can computably enumerate those elements which are *not* the identity. If g is defined as a product of the given generators, we can start enumerating the images of natural numbers under the permutation g. This method would never determine that an element is the identity, but if g is not the identity we know this as soon as g moves some number. So the word problem of any finitely generated subgroup of $\mathcal{C}$ is the *complement* of a computably enumerable set, that is, it is a *co-c.e.* set.

The first result just comes down to considering the left regular representation of a group, so the proof was essentially given by Cayley [Cay] in 1878.

Theorem 1.1. *Every finitely generated group with computable word problem is embeddable in* $\mathcal{C}$.

Proof. We use the *short-lex order* for words from a finitely generated group $G = \langle x_1, \ldots, x_m; R \rangle$. That is, fix a linear ordering of the generators and their inverses. The order on words is then first by length, and then by lexicographical order on words of the same length.

If G has computable word problem we can effectively make a list

$$1 = w_0, w_1, w_2, \ldots$$

of the *distinct* elements of G. Fix an element w_i of G. We can then effectively calculate the index k where $w_i w_j = w_k$ in the above list. Then the permutation corresponding to left multiplication by w_i maps to the permutation $j \to k$ of $\mathbb{N}$.

In short, we can effectively calculate the left regular permutation representation. □

If $G = \langle X; R \rangle$ is a finitely generated group with a computably enumerable set of defining relators then we can enumerate all the consequences of the defining relators so the word problem $WP(G)$ is computably enumerable. If G is embeddable in $\mathcal{C}$ then its word problem is both c.e. and co-c.e and is thus computable by the basic lemma of computability theory discussed above.

Corollary 1.2. *A finitely generated group* G *with a computably enumerable set of defining relators is embeddable in* $\mathcal{C}$ *if and only if* G *has computable word problem.*

So the question is now: Which other finitely generated groups are embeddable in $\mathcal{C}$?

2. Turing degrees and truth-table degrees

We often think in terms of relative computability: "I don't know how to compute B, but if I could compute A then I could compute B". Turing himself [Tur2] formulated the general definition of relative computability in terms of *oracle Turing machines*. Essentially, an oracle Turing machine is an ordinary computer (Turing machine) equipped with a special hardware slot for an *oracle*,

which can hold total information about a set A of words over the input alphabet Σ of the machine. The machine has a special query register and a new branch instruction which says go to different states depending on whether or not the word w in the query register is in the set A or not. Note that an oracle machine still has a fixed finite program so there are still only countably many oracle machines and there is an effective list of all of them. But the oracle can contain information about an arbitrary set. We write M^A to denote the oracle machine M with an oracle for A.

In his fundamental 1944 paper, Emil Post [Pos] realized that oracle machines give the basic foundation for a general theory of computability. A set B is *Turing reducible* to a set A, written $B \leq_T A$, if there is some oracle Turing machine M which, when given the oracle for A, computes B. Two sets A and B are *Turing equivalent*, written $A \equiv_T B$, if $A \leq_T B$ and $B \leq_T A$. This just says that given total information about A, one can compute B, and given total information about B, one can compute A. It is easy to see that $\equiv_T$ is an equivalence relation on the family of all subsets of words over finite alphabets, and equivalence classes are called *Turing degrees*. Since there are only countably many Turing machines and $\mathcal{P}(\Sigma^*)$ has the cardinality of the continuum, there are continuumly many distinct Turing degrees. Turing degrees give a precise definition of when two sets are "computationally equivalent".

The Halting Problem is the set K of all pairs (M, w), where M is a Turing machine and w is a word on the input alphabet of M such that M eventually halts when started with input w. Turing proved that the Halting Problem is not computable in his original paper. Although it is not computable, we can see that the Halting Problem is a c.e. set as follows. We can effectively enumerate all triples (M, w, n) where M is a Turing machine, w is a word on the input alphabet of M and n is a positive integer. When such a triple is enumerated, simulate M on input w for n steps. If M halts during this process, we can enumerate (M, w) as a halting pair. The Halting Problem is the basic example of a computably enumerable set which is not computable.

A Turing degree is a *c.e. degree* if it contains a c.e. set. There are only two "obvious" c.e. Turing degrees: the degree 0 of computable sets, and the degree $0'$ of the Halting Problem. (A c.e. degree may contain many sets which are not c.e. For example, the complement of the Halting Problem, which consists of those pairs (M, w) such that M never halts when started with input w.)

Post raised the basic question of whether or not there are any c.e. Turing degrees besides 0 and $0'$. This question became known as "Post's Problem" and was not settled until 1957 when it was simultaneously resolved by Friedburg and by Muchnik [Soa]. Computability theory has since shown that there

are $\aleph_0$ distinct c.e. Turing degrees. Any set is Turing equivalent to its complement, so the Turing degrees containing co-c.e. sets are exactly the c.e. Turing degrees.

In the paper under discussion, Post introduced a stronger version of computational equivalence, called *truth-table equivalence*. The set B is *truth-table reducible* to the set A, written $B \leq_{tt} A$, if there is an oracle Turing machine M^A and a computable function $\beta : \Sigma^* \to \mathbb{N}$ such that, on input w, M^A decides if $w \in B$ in no more than $\beta(w)$ steps. One can think that, given w, we can compute a loop-free flow chart of size at most $\beta(w)$ for deciding whether or not $w \in B$. Of course, some of the steps may be questions to the oracle for A.

Bounding the number of steps in a computation is now the basis of computational complexity and so is very natural, although we are here working in a very general setting. To clarify the definition it is helpful to consider a Turing reduction which is *not* a truth-table reduction. Suppose that B and C are any two disjoint c.e. sets. We claim that $B \leq_T B \cup C$. To decide if $w \in B$, ask the oracle if $w \in B \cup C$. If not, then $w \notin B$. If yes, start computably enumerating both B and C. Since B and C are disjoint, w will occur in exactly one of the two sets and we then know whether or not it belongs to B. This is not a truth-table reduction since in general there is no computable bound on how far we need to enumerate B and C.

Sets A and B are *truth-table equivalent*, written $A \equiv_{tt} B$, if $A \leq_{tt} B$ and $B \leq_{tt} A$, and equivalence classes are called *truth-table degrees*. Post introduced this notion because he was able to solve Post's problem for truth-table degrees by showing that there is a c.e. set B with $0 <_{tt} B <_{tt} K$. It turns out that the structure of truth-table degrees within a single Turing degree is very rich. Jockusch [Joc] showed that any Turing degree either contains infinitely many truth-table-degrees or is itself a single truth-table degree, and this later case is very rare in a precise sense. Indeed, Degtev [Deg] showed that any nonzero c.e. Turing degree contains infinitely many pairwise incomparable c.e. truth-table degrees.

As with Turing degrees, any set is truth-table equivalent to its complement and a truth-table degree is said to be a c.e. degree if it contains a c.e. set. We have pointed out that the computational complexity of the word problem of a finitely generated group is an invariant of the group and does not depend on a given finitely generated presentation. In discussing the diversity of finitely generated subgroups of $\mathcal{C}$, we need the fact that the truth-table degree of the word problem of a finitely generated group G is an invariant of the group.

3. HNN extensions

A basic construction in infinite group theory is that of taking an HNN extension, introduced by G. Higman, B. Neumann and H. Neumann in 1949. (See [LS] for a detailed discussion of HNN extensions.) Suppose that we have a group $G = \langle X; R\rangle$ and two subgroups A and B of G with an isomorphism $\phi : A \to B$. The corresponding HNN extension is the group

$$H = \langle X, t; R, tat^{-1} = \phi(a), a \in A\rangle.$$

We have added a new generator t, called the *stable letter*, and new relations saying that conjugating an element $a \in A$ by t gives $\phi(a)$. The first basic fact, proved by Higman, Neumann and Neumann, is that G is embedded in H by the map $x \mapsto x$. A non-empty sequence

$$g_0, t^{\epsilon_1}, g_1, \ldots, t^{\epsilon_n}, g_n$$

with the $g_i \in G$ and $\epsilon_i = \pm 1$ is *reduced* if either $n = 0$ (there are no t letters) and g_0 is not the identity in G, or, if $n \geq 1$, then there is no consecutive subsequence t, g_i, t^{-1} with $g_i \in A$ or t^{-1}, g_j, t with $g_j \in B$.

Complete information about the word problem in an HNN extension is given by

Lemma 3.1. *Britton's Lemma. If the sequence* $g_0, t^{\epsilon_1}, g_1, \ldots, t^{\epsilon_n}, g_n$ *is reduced then the product* $g_0 t^{\epsilon_1} g_1 \cdots t^{\epsilon_n} g_n \neq 1$ *in* H.

In general, computation of the word problem for H involves computing the word problem for G, determining membership in the subgroups A and B, and computing the isomorphisms ϕ and ϕ^{-1}. If the subgroups A and B are finitely generated then ϕ and ϕ^{-1} are certainly computable, but we want to take HNN extensions over infinitely generated subgroups and then computability of the isomorphisms also needs to be verified.

Our first application of HNN extensions is the following.

Observation 3.2. For any truth-table degree $\boldsymbol{t}$, there is a two-generator group G with word problem of truth-table degree $\boldsymbol{t}$ which is an HNN extension of the free group of rank 2.

Proof. If $S \subset \{1, 2, \ldots\}$, let

$$G(S) = \langle a, b, t; tat^{-1} = b, tb^i a^i b^i t^{-1} = a^i b^i a^i, i \in S\rangle.$$

The groups $G(S)$ all have the structure of an HNN extension of the free group $F = \langle a, b\rangle$. Any subset of $\{a, b^i a^i b^i : i \geq 1\}$ freely generates a

free subgroup of F since no a can ever be cancelled in a reduced product of the given generators. Similarly for the subset $\{b, a^i b^i a^i : i \geq 1\}$. Let $A = gp\{a, b^i a^i b^i : i \in S\}$ be the subgroup of F generated by the displayed elements and let $B = gp\{b, a^i b^i a^i : i \in S\}$. We have added a new generator t and the relations saying that conjugation by t sends each generator of A to the corresponding generator of B. This map is an isomophism since A and B are freely generated by the displayed sets of generators. Since $G(S)$ is generated by $\{a, b, t\}$ we can use the first relation to eliminate the generator b, seeing that $G(S)$ is a 2-generator group.

We can use Britton's Lemma to show that the word problem of $G(S)$ is truth-table equivalent to membership in the set S. First of all, for any $j \geq 1$,

$$t\ b^j a^j b^j\ t^{-1} = a^j b^j a^j$$

in $G(S)$, that is,

$$t\ b^j a^j b^j\ t^{-1} a^{-j} b^{-j} a^{-j} \in WP(G)$$

if and only if $j \in S$, so $S \leq_{tt} WP(G)$.

Let w be a nonempty freely reduced word on the group alphabet $\{a, b, t\}^{\pm 1}$. If $w = 1$ in $G(S)$, Britton's Lemma says that either w contains no t's and $w = 1$ in the free group or w must contain either a subword tut^{-1} where $u \in A$ or a subword $t^{-1}vt$ where $v \in B$. If w contains such a subword tut^{-1} then we have $tut^{-1} = \phi(u)$ in B. In our particular case, calculating $\phi(u)$ involves simply interchanging the letters a and b. So we can make a *t-reduction* by replacing tut^{-1} by $\phi(u)$, thus reducing the number of t's. Of course, we then freely reduce the result. Exactly the same remark applies if w contains a subword of the form $t^{-1}vt$ where $v \in B$.

But deciding if a reduced word of the free group $\langle a, b \rangle$ is in A or B can be done in linear time *given* membership in S. Since no a is ever cancelled in a reduced product of the generators of A, given a reduced product

$$a^{i_1} b^{j_1} a^{i_2} b^{j_2} \ldots a^{i_k} b^{j_k} a^{j_k}$$

we can calculate if the exponents on the occurrences of a and b are correct for membership in A or not, and similarly for B. So knowing membership in S allows us to reduce an arbitrary word w in quadratic time. We either arrive at a nonempty word in which no reductions are possible, so $w \neq 1$ in $G(S)$, or we arrive at the empty word and $w = 1$ in $G(S)$. Since we can compute the number of possible steps in such a reduction from w, this is a truth-table reduction and we have concluded that $WP(G(S)) \leq_{tt} S$. Thus the word problem for $G(S)$ and S are truth-table equivalent. □

Actually, the reduction above and the reductions which we later consider are much stronger than general truth-table reductions. Since we are dealing with both words in group generators and natural numbers, for measuring the size of inputs we consider natural numbers as written in unary. Then the functions bounding the number of steps in our reductions will be a single exponential or a polynomial function of the length of inputs, which is the desired condition in complexity theory. Since we are dealing with oracles for non-computable sets, we do not stress this fact but simply point out that it is indeed the case.

Observation 3.3. For a group $G(S)$ as above, if S is a co-c.e. set, then $WP(G)$ is also co-c.e.

Proof. As a first example of what we want to do, consider a word w with only one possible reduction to the identity, say

$$b^{-20}a^{-20}b^{-20}\ t^{-1}\ a^{20}b^{20}a^{20}\ a^{30}b^{30}a^{30}\ t\ b^{-30}a^{-30}b^{-30}$$
$$|$$
$$20 \in S?$$
$$30 \in S?$$
$$|$$
$$1.$$

Now $w \neq 1$ in $G(S)$ only if $20 \notin S$ or $30 \notin S$. Since we can enumerate the complement $\overline{S}$ of S, if either 20 or 30 is not in S this fact will appear in our enumeration and we can then enumerate w as a word not equal to the identity.

In general, for each freely reduced word w we can construct the tree of all possible reductions of w. On each edge write the exponents needed to be in S in order to make the reduction. The reduction is *not* possible exactly if some exponent in *not* in S. Since we can enumerate the complement $\overline{S}$ of S, we will eventually discover any reduction which is not possible and cancel that branch. Of course we cancel any branch that ends in a nonempty reduced word to which no reductions could possibly be applied. A word w is equal to 1 in $G(S)$ if and only if some some branch is not eventually canceled. So enumerating each word with all branches canceled exactly enumerates the complement of the word problem. □

This method also shows that the word problem is generically easy for all the groups $G(S)$. By this we mean that there is a partial algorithm which is always correct and which works on a set of asymptotic density 1 (see [KMSS]). Most reduction trees show that the starting word cannot possibly reduce to 1. In particular, a freely reduced word w can equal the identity in $G(S)$ only if the exponent sum on t in w is exactly 0.

The probability that a random word of length n has the exponent sum on t exactly equal to 0 goes to 0 as $n \to \infty$. So a linear time generic algorithm for the word problem of $G(S)$ is to check if the exponent sum on t in w is 0. If not, answer $w \neq 1$, and if the exponent sum is 0 do not answer. Note that the same generic algorithm works for all the groups $G(S)$. Indeed, it works for all HNN extensions of finitely generated groups.

We have seen that there are 2-generator groups of every possible truth-table degree and indeed, there are 2-generator groups with co-c.e. word problem of every possible c.e. truth-table degree. It is not at all clear if any of the groups $G(S)$ just considered are embeddable in $\mathcal{C}$.

Higman, Neumann and Neumann introduced HNN extensions to prove, among other things, that every countable group can be embedded in a 2-generator group. After working through the example of the groups $G(S)$ we see that the "standard" HNN embedding of a given presentation of a countable group C into a two generator group preserves the truth-table degree of the word problem. (Note that the computational complexity of the word problem is *not* an invariant of the group for infinitely generated presentations. Indeed, is easy to write down an infinitely generated presentation of the cyclic group of order 2 for which the word problem is not computable.)

Observation 3.4. Let $G = \langle X; R \rangle$ be a group where the set X of generators is either finite or is the infinite set $\{x_1, \ldots, x_n, \ldots\}$ where the indices range over $\mathbb{N}^+$. The standard HNN embedding of G into a 2-generator group H preserves the truth-table degree of the word problem of the given presentation of G.

Proof. We first take the free product $G * F$ where $F = \langle a, b \rangle$ is a free group on two generators, and then take the HNN extension

$$H = \langle G * F, t; tat^{-1} = b, \qquad tb^i a^i b^i t^{-1} = x_i a^i b^i a^i, i \geq 1 \rangle.$$

The relations are exactly the same as the relations we used in the groups $G(S)$ except that the generator x_i appears once in the i-th relation involving conjugation by t. Now H is a 2-generator group since the first relation can be used to express a in terms of t and b, and the i-th relation can be used to express x_i in terms of a, b and t. And G is embedded in H by the basic property of HNN extensions.

We need to check that $WP(H) \equiv_{tt} WP(G)$. First, $WP(G) \leq_{tt} WP(H)$ since G is embedded in H by the map $x_i \to x_i$. As before, the subgroups $A = gp\{a, b^i a^i b^i\}$ and $B = gp\{b, x_i a^i b^i a^i\}$ are freely generated by the displayed sets of generators because no a is ever cancelled in a reduced product of generators from the first set and no b is ever cancelled in a reduced product of generators from the second set. To see if a word w is in one of the subgroups we can determine the product of generators needed from reading w. For example,

$$g_1 a^5 b^5 g_2 a^2 b^{-3} a^{-3} g_3 \in B$$

if and only if $g_1 = x_5$ in G and both g_2 and g_3 are equal to the identity in G. The map sending a generator of A to the corresponding generator of B, or vice versa, again simply interchanges a and b and deletes or inserts an x_i. We can again test all possible t-reductions and $WP(H) \leq_{tt} WP(G)$. □

4. Finitely generated groups embeddable and not embeddable in $\mathcal{C}$

Nies and Sorbi [NS] have shown that for every c.e. truth-table degree there is 3-generator subgroup of $\mathcal{C}$ with word problem of that degree. We give here a simple construction for 2-generator subgroups of $\mathcal{C}$.

Theorem 4.1. *For every nonzero c.e. truth-table degree* $\mathbf{t}$ *there is a* 2*-generator subgroup of* $\mathcal{C}$ *with word problem of degree* $\mathbf{t}$.

Proof. Let $B = \{b_{i,j}, i \in \mathbb{N}, j \in \mathbb{Z}\}$ be a set of "blue" elements and let $R = \{r_{i,j}, i \in \mathbb{N}, j \in \mathbb{Z}\}$ and $Y = \{y_{i,j}, i \in \mathbb{N}, j \in \mathbb{Z}\}$ be respectively disjoint sets of "red" elements and "yellow" elements. Let $P = B \sqcup R \sqcup Y$ be their disjoint union. There is a computable bijection from P to $\mathbb{N}$ so we can consider permutations of P. We think of P as initially arranged in rows

$$\cdots, b_{i,-1}, r_{i,-1}, y_{i,-1}, \qquad b_{i,0}, r_{i,0}, y_{i,0}, \qquad b_{i,1}, r_{i,1}, y_{i,1}, \cdots$$

of triples, where the i-th row consists of all elements with first index i.

Let A be any infinite c.e. subset of $\mathbb{N}$ with $0 \notin A$ and let $f : \mathbb{N} \to A$ be a computable bijection.

Define computable permutations σ and β as follows:

(1) $\sigma(b_{i,j}) = b_{i,j+1}$, $\sigma(r_{i,j}) = r_{i,j+1}$ and $\sigma(y_{i,j}) = y_{i,j+1}$. Thus σ is the shift map on the $\mathbb{Z}$ indices.

(2) $\beta = \prod_{i=0}^{\infty} (b_{i,0}, y_{i,0})(b_{i,f(i)}, r_{i,f(i)})$ (An infinite product of 2-cycles.)

For each i, β interchanges the elements $b_{i,0}$ and $y_{i,0}$ and also interchanges $b_{i,f(i)}$ and $r_{i,f(i)}$.

Both σ and β are computable since f is computable. Note that neither σ nor β ever move an element out of its row, that is, they never change an i index.

We need to show that the word problem for the subgroup G generated by σ and β has the same truth-table degree as A.

Claim 4.1. For any $m \in \mathbb{N}$, the permutations β and $\sigma^{-m} \beta \sigma^m$ commute if and only if $m \notin A =$ range f. Thus $A \leq_{tt} WP(G)$.

Proof. That β and $\sigma^{-m}\beta\sigma^m$ commute means exactly that $\pi_1 = \beta\sigma^{-m}\beta\sigma^m$ and $\pi_2 = \sigma^{-m}\beta\sigma^m\beta$ are the same permutation.

Suppose that $m > 0$ is not in A, that is, m is not equal to $f(i)$ for any i. By successively applying the permutations in π_1 and π_2, check that both π_1 and π_2 fix all triples $(b_{i,j}, r_{i,j}, y_{i,j})$ where j is not equal to one of the values $0, -m, f(i)$ or $f(i) - m$. Now check that both π_1 and π_2 send $(b_{i,0}, r_{i,0}, y_{i,0})$ to $(y_{i,0}, b_{i,0}, r_{i,0})$ and $(b_{i,-m}, r_{i,-m}, y_{i,-m})$ to $(y_{i,-m}, r_{i,-m}, b_{i,m})$, while sending $(b_{i,f(i)}, r_{i,f(i)}, y_{i,f(i)})$ to $(r_{i,f(i)}, b_{i,f)i)}, y_{i,f(i)})$ and $(b_{i,f(i)-m}, r_{i,f(i)-m}, y_{i,f(i)-m})$ to $(r_{i,f(i)-m}, b_{i,f(i)-m}, y_{i,f(i)-m})$.

If $m = f(i')$ for some i', just check that π_1 sends $(b_{i,0}, r_{i,0}, y_{i,0})$ to $(r_{i,0}, y_{i,0}, b_{i,0})$ while π_2 sends $(b_{i,0}, r_{i,0}, y_{i,0})$ to $(y_{i,0}, b_{i,0}, r_{i,0})$ so β and $\sigma^{-m}\beta\sigma^m$ do not commute. □

To solve the word problem for G given A as an oracle, first note that, since β has order 2, given a word w in the generators of G, we can effectively reduce w to an equivalent word v of the form

$$\sigma^{m_0}\beta\sigma^{m_1}\beta\cdots\sigma^{m_{k-1}}\beta\sigma^{m_k}$$

by successively combining adjacent powers of σ or β, deleting even powers of β, and replacing odd powers of β by β.

Since σ is the shift map on $\mathbb{Z}$ indices, v cannot equal the identity unless the sum of the exponents m_i equals 0. If the exponent sum is 0, let m be the sum of the positive exponents on σ occurring in v. We use the oracle to determine which of the numbers $1, \ldots, 2m$ are in A. It is important to note that we only need to know whether or not these numbers are in A and we do *not* need to know the values of i for which $f(i)$ is one of the given numbers. Finding such a value of i would require an unbounded search and would not give a truth-table reduction.

The first observation is that if i and i' are any two indices with $f(i) > 2m$ and $f(i') > 2m$, then v acts as the identity on row i if and only if v acts as the identity on row i'. This is because v can only interchange elements with subscripts differing from 0 or $f(i)$ by at most m, and since $f(i) > 2m$, these exchanges cannot interact. The action of v on triples with j in the interval, mirrors the action of v on elements with j in the interval $[-m, m]$. If v is not the identity on the elements $(b_{i,j}, r_{i,j}, y_{i,j})$ with $j \in [-m, m]$ or with $j \in [f(i)-m, f(i)+m]$ then v is not the identity.

If we need to check further then, for each number $l, 1 \leq l \leq 2m$ which is in A, suppose that $f(i) = l$ and check the effect of v on elements with second indices ranging from $(i, -m)$ to $(i, l + m)$. Then v is the identity in G if and only if it acts as the identity on each of these intervals. We have specified a

bounded, indeed polynomially bounded, number of elements to check, so we have $WP(G) \leq_{tt} A$. □

If G and H are groups which are both embeddable in $\mathcal{C}$ then their direct product $G \times H$ is also embeddable in $\mathcal{C}$. (There is a bijection $\mathbb{N} \to \mathbb{N} \times \mathbb{N}$ and $G \times H$ permutes $\mathbb{N} \times \mathbb{N}$ by letting G permute the first factor and H permute the second factor.) Taking the direct product of two-generator examples with a free group of finite rank r shows that there are k-generator subgroups of $\mathcal{C}$ with word problem of truth-table degree $\boldsymbol{t}$ for all $k \geq 2$.

Morozov [Mor] constructed two-generator groups with co-c.e. word problem which are *not* embeddable in $\mathcal{C}$, answering a question posed by Graham Higman. We show here that this construction preserves truth-table degree, so we have

Theorem 4.2. *For every nonzero c.e. truth-table degree* $\boldsymbol{t}$*, there is a two-generator group with word problem of degree* $\boldsymbol{t}$ *which is not embbedable in* $\mathcal{C}$.

Proof. Let A be a c.e. set of truth-table degree $\boldsymbol{t}$ and let

$$P = \bigoplus_{n \geq 1} \langle x_n ; {x_n}^{p_n} \rangle$$

where p_n is equal to 2 if $n \notin A$ and otherwise p_n is the least prime greater than $2s$ if n appears in A at step s of the enumeration of A. This is a direct sum of finite cyclic groups of prime order and so all the generators x_i commute.

We note that P itself is embeddable in $\mathcal{C}$. Indeed, any countable, locally finite group G is embeddable in $\mathcal{C}$. That G is locally finite means that every finitely generated subgroup of G is finite. Fix an enumeration $g_1, g_2, \ldots$ of the non-identity elements of G. Let $G_1 = \langle g_1 \rangle$ be the subgroup of G generated by g_1. Let I_1 be the initial segment of $\mathbb{N}$ of cardinality $|G_1|$. Fix a bijection η_1 from G_1 to I_1 which sends the identity element to 0 and let ρ_1 be the induced permutation representation of G_1 on I_1. If G_{n-1} has already been defined, let g_{i_n} be the first element in the enumeration of G which is not in G_{n-1} and let $G_n = \langle G_{n-1}, g_{i_n} \rangle$. Let I_n be the initial segment of $\mathbb{N}$ of cardinality $|G_n|$ and choose a bijection η_n from G_n to I_n which extends η_{n-1}, so the induced permutation representation of G_n on I_n extends ρ_{n-1}. Define $\Psi : G \to \mathcal{C}$ as follows. If $g_i \in G$, let G_k be the first group for which $g_i \in G_k$. Then $\Psi(g_i)$ is the permutation of $\mathbb{N}$ which agrees with ρ_k on I_k and fixes all elements of $\mathbb{N}$ outside of I_k. For every g_i, $\Psi(g_i)$ moves only finitely many elements so Ψ is an embedding of G into the subgroup of finitary permutations and all finitary permutations are computable.

Returning to the group P as given, we need to check that the word problem for P has the same truth-table degree as A. First, ${x_j}^2 = 1$ in P if and only if

$j \notin A$ so $A \leq_{tt} WP(P)$. Since all the generators commute, we can write any element in the form

$$w = \quad {x_{j_1}}^{m_1} \ldots {x_{j_r}}^{m_r} \neq 1.$$

Such a product is not the identity if and only if some ${x_j}^{m_j} \neq 1$.

Now ${x_j}^k \neq 1$ if and only if ($j \notin A$ and k is odd) OR ($j \in A$ and $p_j \nmid k$). So first, for each j, ask the oracle if $j \in A$. For those $j \notin A$, we just check if m_j is odd, and if so, $w \neq 1$ in P. If this does not determine the status of w, let m be the maximum of all the exponents m_j in w and set $s = [\frac{m}{2}]$. Enumerate A for s steps. If j appears in A in this process, we know the value of p_j and can thus check if $p_j \mid m_j$. If $j \in A$ but has not appeared in s steps of the enumeration then $p_j > m$ and so ${x_j}^{p_j} \neq 1$ in P. We have bounded the number of steps in this computation from the form of w so this is indeed a truth-table reduction and $WP(P) \leq_{tt} A$ and therefore we have $WP(P) \equiv_{tt} A$.

We can now do the standard HNN embedding of the group $P = \langle x_1, \ldots; R\rangle$ into a two-generator group. Let

$$H(P) = \left\langle P * \langle a, b\rangle, t; tat^{-1} = b, t \quad x_i b^i a^i b^i \quad t^{-1} = a^i b^i a^i, i \geq 1\right\rangle.$$

The Observation on the standard HNN embedding into a two-generator group shows that the word problem for $H(P)$ is truth-table equivalent to A.

Since $H(P)$ has two generators, an embedding $\varphi : H(P) \to \mathcal{C}$ would be effectively calculable. In this case, $\{\varphi(x_i)\}$ would be a computable family of computable permutations. We write $o(\pi)$ for the order of a permutation π. We now observe that if $\{\pi_i\}$ is a computable family of computable permutations of $\mathbb{N}$, all of which have prime order, then the function $i \to o(\pi_i)$ is computable. To check this, first start enumerating elements of $\mathbb{N}$ until we find an m with $\pi_i(m) \neq m$. Now enumerate the images ${\pi_i}^j(m), j = 1, \ldots, k$ where the first repetition is the k-th. Then k must be the order of π_i since the order is a prime. But $\varphi(x_n)$ has order 2 if and only if $n \notin A$, so we conclude that $H(P)$ cannot be embeddable in $\mathcal{C}$. □

5. Other groups of computable automorphisms

Consider the group $\mathcal{A}$ of computable automorphisms of the free group $F = \langle y_1, y_2, \ldots\rangle$ of rank $\aleph_0$. Then $\mathcal{C}$ is embeddable in $\mathcal{A}$ by just permuting the generators. To see that $\mathcal{A}$ is embeddable in $\mathcal{C}$ we consider computable permutations of $\mathbb{N}^2$.

List the nonempty words of F in the following way. Fix the lexicographic order

$$y_1 < {y_1}^{-1} < \ldots < y_k < {y_k}^{-1} < \ldots$$

on the set Y of generators and their inverses. Let LL_k consist of all freely reduced words on $y_1, \ldots, y_k$, listed in shortlex order, which contain y_k or its inverse and no generators with higher subscript. Let $w_{i,k}$ be the i-th word in list LL_k. If α is a computable automorphism of F, let $\alpha(w_{i,k}) = w_{r,s}$ then α corresponds to the permutation which sends 0 to 0 and $(i,k) \to (r,s)$.

Since each of $\mathcal{C}$ and $\mathcal{A}$ is embeddable in the other, up to isomorphism they have the same set of subgroups. To see that $\mathcal{C}$ and $\mathcal{A}$ are not isomorphic we consider normal subgroups. Schreier and Ulam [SU] investigated the group $\mathcal{S}$ of *all* permutations of $\mathbb{N}$ in 1933. They showed that the only normal subgroups of $\mathcal{S}$ other than $\{1\}$ and the whole group are S_∞, the group of all finitary permutations of $\mathbb{N}$, and A_∞, the group of alternating finitary permutations. Note that these last two groups are torsion groups where all elements have finite order. Clement Kent [Ken] showed that the normal subgroup structure of $\mathcal{C}$ is the same as that of $\mathcal{S}$. The group of inner automophisms of F is a normal subgroup of $\mathcal{A}$ and is torsion-free. So $\mathcal{C}$ and $\mathcal{A}$ are not isomorphic.

Similarly, we have mutual embeddability with $\mathcal{C}$ for many other groups of computable automorphisms. For example, the group of computable automorphisms of the free abelian group of rank $\aleph_0$ or the computable automorphisms of the direct sum of $\aleph_0$ cyclic groups of order p for p a fixed prime.

6. Groups with a co-c.e. presentation

The following observation is well-known.

Observation 6.1. A finitely generated group $G = \langle X; R \rangle$ where the set R of defining relators is computably enumerable has a presentation with a computable set of defining relators.

Proof. Since R is computably enumerable, there is a computable function $f : \mathbb{N}^+ \to R$ which is onto R. We check that G is isomorphic to

$$\langle\ X, z; z, z^i f(i)\ \rangle$$

which is a computable presentation. This is a presentation of G since we have just added a new generator and set it equal to the identity. So the relations are equivalent to $\{f(i), i \in \mathbb{N}^+\}$ which is just R. This second presentation is computable since to check if a word of the form $z^i w$ is a defining relation we just compute $f(i)$ and see if it is w. □

This raises the natural question of what one can say about groups which have a co-c.e. presentation. The situation now becomes very different from that of c.e. presentations. To explain the result we need to introduce some more computability theory.

The original complexity hierarchy is of course the Borel Hierarchy [Bor]. The corresponding hierarchy in computability theory is the Arithmetic Hierarchy. We list here a few basic facts but see Soare [Soa] for a detailed discussion. Since we are considering words in groups, we consider subsets of Σ^*, the set of all words on a nonempty finite alphabet Γ. The basic family of sets is the family Δ_1 of all computable subsets of Γ^*. The family Σ_1 of all c.e. subsets of Γ^* is the family of all subsets of the form

$$\{x : \exists \overline{y} R(x, \overline{y})\}$$

where $\overline{y}$ is a sequence $y_i, \ldots, y_l$ of variables and $R(x, \overline{y})$ is an arbitrary computable relation. The family Π_1 of co-c.e. sets is the family of subsets of the form

$$\{x : \forall \overline{y} R(x, \overline{y})\}.$$

Note that $\Delta_1 = \Sigma_1 \cap \Pi_1$ by the basic lemma of computability theory.

The hierarchy is continued by alternating quantifiers. Thus the family Σ_2 is the family of all subsets of the form

$$\{x : \exists \overline{y} \forall \overline{z} R(x, \overline{y}, \overline{z})\}$$

where where $R(x, \overline{y}, \overline{z})$ is a computable relation and the family Π_2 of is the family of subsets of the form

$$\{x : \forall \overline{y} \exists \overline{z} R(x, \overline{y}, \overline{z})\}.$$

By definition, $\Delta_2 = \Sigma_2 \cap \Pi_2$.

K', the *jump* of the Halting Problem K, is Halting Problem for oracle Turing machines with an oracle for K. The Turing degree of K' is denoted by $0''$. Post's Theorem shows that Σ_2 is the family of sets enumerable by Turing machines with an oracle for K and Δ_2 is the class of sets computable by Turing machines with an oracle for K.

The Arithmetic Hierarchy is continued by repeatedly alternating quantifiers and the sets obtained are all distinct. A remarkable theorem about Δ_2 is the Shoenfield Limit Lemma: A set $S \in \Delta_2$ if and only if there is a computable function $f : \mathbb{N} \times \Sigma^* \to \{0, 1\}$ such that $w \in S$ if and only if $lim_{n \to \infty} f(n, w) = 1$. We can think that at each stage n, f is trying to determine whether or not $w \in S$, $(f(n, w) = 1)$ or $w \notin S$, $(f(n, w) = 0)$ and for each w the value is eventually correct.

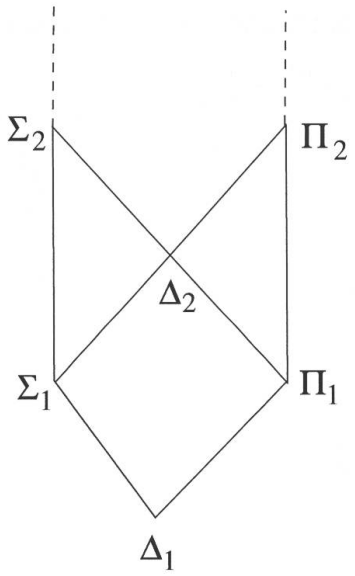

FIGURE 1
The Arithmetic Hierarchy

Note that a finitely generated group with a Σ_2 set of defining relators has a presentation with a Δ_2 set of defining relators. The proof is the same as before by replacing the word "computable" by "computable from an oracle for the Halting Problem". The following theorem arose in a conversation with Carl Jockusch during the preparation of this paper.

Theorem 6.1 (Jockusch). *A finitely generated group G with a Σ_2 set of defining relators has presentation with a co-c.e. set of defining relators.*

Proof. We can suppose that we have a Δ_2 presentation $G = \langle X; R\rangle$ for G. The Shoenfield Limit Lemma says that a set R is Δ_2 if and only if there is a computable $\{0, 1\}$-valued function $f(s, u)$ of two variables such that $w \in R$ if and only if $\lim_{s\to\infty} f(s, w) = 1$.

Given the above Δ_2 presentation and f, let

$$G = \langle X, z; z, z^s w\rangle$$

such that $f(t, w) = 1$ for all $t \geq s$.

First, the above presentation is indeed a presentation of the given group G. Since $z = 1$ in G, there is a relation $z^s w$ setting $w = 1$ in G exactly if $f(t, w) = 1$ for all t greater than or equal to some s. By the property of the function f, this happens if and only if $w \in R$. We just have to check that we can enumerate the *complement* of the displayed relators. We begin listing all words on the generators $X \cup \{z\}$ and their inverses in short-lex order and start listing

all pairs (n, w). We never enumerate z. If a word u does not have the form $z^s w, s \geq 1$, we enumerate u. Otherwise, if $f(s, w) = 1$ we do not enumerate u. If $f(s, w) = 0$, we enumerate *all* words $z^j w$ with $1 \leq j \leq s$. By the property of f, we enumerate exactly the words which do not appear in the above presentation of G. □

References

[Bor] E. Borel, *Leçons sur la théorie des fonctions*. Gauther-Villars, Paris, 1898. JFM 29.0336.01

[Cay] A. Cayley, On the theory of groups. *Proc. London Math. Soc.* **9** (1878), 126–133. JFM 10.0104.01

[Deg] A. N. Degtev, Hereditary sets and tabular reducibility. *Algebra i Logika* **11** (1972), 257–269. Zbl 0283.02035 MR 0313033

[Joc] C. G. Jockusch, Relationships between reducibilities. *Trans. Amer. Math. Soc.* **142** (1969), 229–237. Zbl 0188.02604 MR 0245439

[KMSS] I. Kapovich, A. Myasnikov, P. Schupp, and V. Shpilrain, Generic-case complexity, decision problems in group theory and random walks. *J. Algebra* **264** (2003), 665–694. Zbl 1041.20021 MR 1981427

[Ken] C. Kent, Constructive analogues of the group of permutations of the natural numbers. *Trans. Amer. Math. Soc.* **104** (1962), 347–362. Zbl 0105.24802 MR 0140406

[Kle] S. C. Kleene, Recursive predicates and quantifiers. *Trans. Amer. Math. Soc.* **53** (1943), 41–73. Zbl 0063.03259 MR 0007371

[LS] R. C. Lyndon and P. E. Schupp, *Combinatorial Group Theory*. Classics in Mathematics, Springer, 2001 Zbl 0997.20037 MR 1812024

[Mor] A. S. Morozov, Once again on the Higman question. *Algebra i Logika* **39** (2000), 134–144.

[NS] A. Nies and A. Sorbi, Calibrating word problems of groups via the complexity of equivalence relations. *Math. Structures Comput. Sci.* **28** (2018), 457–471. MR 3778212

[Pos] E. L. Post, Recursively enumerable sets of positive integers and their decision problems. *Bull. Amer. Math. Soc.* **50** (1944), 284–316. Zbl 0063.06328 MR 0010514

[SU] O. Schreier and S. Ulam, Über die Permutationsgruppe der natürlichen Zahlenfolge. *Studia Math.* **4** (1933), 134–141. Zbl 0008.20003

[Soa] R. I. Soare, *Turing Computability*. Springer, 2016. Zbl 1350.03001 MR 3496974

[Tur1] A. M. Turing, On computable numbers with an application to the Entscheidungsproblem. *Proc. London Math. Soc.* **42** (1936), 230–265. Zbl 0016.09701 MR 1577030

[Tur2] —— Systems of logic based on ordinals. *Proc. London Math. Soc.* **45** (1939), 161–228. Zbl 0021.09704 MR 1576807

(*Reçu le 11 février 2018*)

Andrey MOROZOV, Sobolev Institute of Mathematics,
Koptyug Prosp. 4, Novosibirsk 630090, Russia

e-mail: morozov@math.nsc.ru

Paul SCHUPP, Department of Mathematics, University of Illinois
at Urbana-Champaign, 1409 West Green Street, Urbana, IL 61801, USA

e-mail: schupp@math.uiuc.edu