HAL open science

REMARKS AND PROBLEMS ABOUT ALGORITHMIC DESCRIPTIONS OF GROUPS

Emmanuel Rauzy

To cite this version:

Emmanuel Rauzy. REMARKS AND PROBLEMS ABOUT ALGORITHMIC DESCRIPTIONS OF GROUPS. 2021. hal-03410992

HAL Id: hal-03410992

https://hal.science/hal-03410992v1

Preprint submitted on 1 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

REMARKS AND PROBLEMS ABOUT ALGORITHMIC DESCRIPTIONS OF GROUPS

EMMANUEL RAUZY

Abstract. We make the case that the so called "global decision problems" should not be investigated solely for groups described by finite presentations. We propose to use descriptions that be algorithms that perform some given tasks, and that encode the considered groups. We motivate this by establishing undecidability results for groups described by recursive presentations, strong enough to prevent an interesting theory of decision problems based on generic recursive presentations to be developed. More importantly, we give an algorithmic characterization of finitely presented groups, in terms of existence of a "marked quotient algorithm" which recognizes the quotients of the considered group. This new point of view leads us to proposing several open questions and directions of research, and much of this paper consists in exposing problems that arise from our first results. Finally, note that we set our study in the category of marked groups, we explain why this is beneficial, and give open questions that arise from the study of decision problems for marked groups.

Contents

Introduction 2
Decision problems for marked groups 8
1. Recursive and finite presentations 10
1.1. Solving decision problems from recursive presentations 10
1.2. The marked quotient algorithm and finitely presented groups 12
1.3. Three more unsolvability results 14
2. Additional discussion 16
2.1. Recursive presentations 16
2.2. The Kreisel-Lacombe-Schoenfield-Ceitin Theorem 18
2.3. Practical implementation 19
2.4. The formalism of [GMW12] 20
3. Recognizable groups 21
3.1. From finite presentations 21
3.2. From finite presentations together with word problem algorithms 23
4. Relative marked quotient algorithms 24
4.1. Definitions and Basic properties 24
4.2. Extending classical uses of marked quotient algorithms to relative algorithms 26
4.3. Group varieties 28
4.4. Elementary marked quotient algorithms 28
References 30

Date: 12/10/2021.

Introduction

The study of decision problems for groups began with Max Dehn who, in 1911 ([Deh11]), formulated the three famous problems which are now associated with his name: the word problem, the conjugacy problem and the isomorphism problem.

His motivation in introducing those came from topology, in particular from the study of the fundamental group, which had been introduced not long before by Poincaré. Because of this, he defined those problems only for finitely presented groups; his 1911 article starts by stating the following (we quote Stillwell's translation, [Deh87]):

> "The general discontinuous groups is given by n generators and m relations between them:
>
> $$R_1(s_{i_1}, ...) = 1$$
> $$.....$$
> $$R_m(s_{i_m}, ...) = 1$$
>
> as first defined by Dyck ([Dyc82, Dyc83]). The results of those works, however, relate essentially to finite groups. The general theory of groups defined in this way at present appears very undeveloped in the infinite case. Here there are above all three fundamental problems whose solution is very difficult and which will not be possible without a penetrating study of the subject."

Dehn then proceeds to defining his three problems.

This extract shows that Dehn introduced what would be known as computability theory (once it was invented) in the field of group theory at the express purpose of studying finitely presented groups, in order to build a theory of finitely presented groups.

If we consider this, in light of the fact that the finitely presentable groups are exactly the fundamental groups of closed manifolds, we can see that from its very beginning the theory of decision problems for groups lay at the intersection, not only of group theory and computability theory, but of group theory, computability theory *and* topology. This is for instance detailed in [Bri02].

The intersection of these three domains has now been well studied, and it is very satisfying to think that there are many instances where a problem which was posed in one field was solved thanks to methods of another one.

Consider now the following question, which could be taken as the starting point of our investigation: *what would the theory of decision problems for groups look like, had it been invented by someone with no knowledge of topology?*

Actually, the first step in the making of an algorithmic theory of groups is not just a thought exercise that we propose here, it was actually written, independently by Malcev in [Mal61] (a translation is available in [Mal71]) and by Rabin in [Rab60]. They defined what is known as a computable group, and this notion can easily be seen to correspond to having solvable word problem with respect to a countably infinite generating set.

Although both of these papers prove some fundamental results, the working frame they propose is not entirely satisfactory: all the notions that those papers introduce were already known, or already expressible, in the language of the existing theory of decision problems for finitely presented groups. And, what's more, as the concept of "computable group" does not encompass all finitely presented groups, because of the Novikov-Boone Theorem, at the time of their writing, the set of groups that admit a finite description that can be manipulated by a computer was already known to exceed that of computable groups.

Thus upon reading those two articles, one might have the impression that, paradoxically, the intersection of group theory and computability theory is strictly contained in the intersection of group theory, computability theory and topology. In this article, we show that this does not have to be the case.

There is another approach to our motivating question which seems to be worth mentioning, and which consists in making analogies with the theory of computable analysis, which is the domain that lies at the intersection of computability theory and analysis.

In this regard, the introduction of groups with solvable word problem naturally corresponds to that of computable reals, decidable and semi-decidable properties of groups could have been defined by analogy with computable functions, etc.

This analogy also suggests that recursively presented groups would also have been introduced, as those are the equivalent of lower (or upper) semi-computable reals, which are well known objects in computable analysis ([Spe49]). Note furthermore that those groups could have thus been defined with no reference to group presentations: a finitely generated group G together with a generating family S admits a recursive presentation on S if and only if there exists an algorithm that recognizes the words in the generators and their inverses that define the identity element of the group, thus solving "half" of the word problem. Throughout this paper, we will usually favor the use of such algorithms over that of recursive presentations. They are called *r.e. algorithm*s, for *recursive enumeration algorithm*s. Recursively presented groups will be called r.p. groups for short. We use the notation $\mathcal{A}_{re}^{G;S}$ for a r.e. algorithm of the group G generated by S, and omit both superscripts G and S when possible. We discuss why we choose to use r.e. algorithms over recursive presentations in Section 2.

The parallel with computable analysis seems fruitful in several regards. For instance, in Section 2, we ask for an equivalent to one of the most important theorems of computable analysis, the Kreisel-Lacombe-Schoenfield-Ceitin Theorem.

Let us come back to our thought exercise, of describing a theory of decision problems for groups, had it been introduced with no topology in mind.

If we are to describe such a theory, one of our main concerns will be to define decision problems for groups that are not necessarily finitely presented. A key remark here is that we must distinguish between the so called "local" and "global" decision problems. This distinction is fundamental, so much so that Miller's well known 1992 survey article ([Mil92]) starts with the following:

> "This is a survey of decision problems for groups, that is of algorithms for answering various questions about groups and their elements. The general objective of this area can be formulated as follows:
> Objective:
> To determine the existence and nature of algorithms which decide
> - local properties - whether or not elements of a group have certain properties or relationships;
> - global properties - whether or not groups as a whole possess certain properties or relationships.
>
> The groups in question are assumed to be given by finite presentations or in some other explicit manner."

Notice that, as opposed to Dehn, Miller allows descriptions of groups that are not finite presentations.

The reason why we introduce this distinction here is the following: the investigation of decision problems in regards to local properties of groups is not affected by whether the considered groups are finitely presented or not -a solution to the word problem in some group depends only on a choice of a generating set for this group, (of a marking of this group), but not of a presentation. We quote here Miller's definition of the word problem:

> "Word problem: Let G be a group given by a finite presentation. Does there exist an algorithm to determine of an arbitrary word w in the generators of G whether or not $w = I$?"

Notice how the assumption that G is given by a finite presentation can be dropped, the resulting definition stays valid in any countable group with a fixed generating family.

Let us now quote Miller's definition of the isomorphism problem:

> "Isomorphism problem: Does there exist an algorithm to determine of an arbitrary pair of finite presentations whether or not the groups they present are isomorphic?"

Here, if one were to take off the terms "finite presentations" from the formulation of this problem, and replace them by "a pair of groups", the problem stops being posed well, since of course an algorithm cannot take as input the abstract mathematical object that is a group.

Thus we must find an adequate replacement for "finite presentations" in this definition, which should also be more general than the notion of computable group, as we have already explained that this notion was too restrictive.

There is one last obvious notion one could use in the hope of finding a unique definition for decision problems about groups that need not be finitely presented: the notion of recursive presentation.

Of course, since a recursive presentation can be encoded in a Turing machine, and since Turing machines can run other Turing machines, recursive presentations can be used to define global decision problems.

However, we have the following proposition, which motivated our writing of this article:

Proposition 1. *Let G and H be finitely generated, recursively presented groups, and suppose that H is a strict quotient of G. Then, the problem of deciding whether a given pair of recursive presentations, that both define either G or H, define the same group, is unsolvable.*

This proposition is stated and proved in Section 1.

This proposition prevents one from building an interesting theory of algorithmic problems for groups described by recursive presentations, and this, not because it shows that the isomorphism problem is unsolvable for groups described by recursive presentations, this is to be expected, but because it is unsolvable in any class of finitely generated groups that contains a group and a strict quotient of it.

An interesting theory of decision problems for groups strives on the fact that most decision problems are unsolvable in general, but become solvable in restricted classes of groups. The aim of the study of decision problems for finitely presented groups could be summed up by the following: "to quantify the complexity of classes of finitely presented groups thanks to solvability and unsolvability of different decision problems".

In that respect, the Adian-Rabin Theorem, which implies unsolvability of a wide range of problems in the class of all finitely presented groups, plays an important role in rendering this program possible.

But it is also fundamental for this program to make sense that there exist classes of groups for which decision problems be solvable, and we can quote here the great achievements that are the solutions to the isomorphism problems for polycyclic groups, for hyperbolic groups, etc. Proposition 1 is a much too powerful equivalent of the Adian-Rabin Theorem, and it kills the study of decision problems for groups described by recursive presentations.

This proposition (and its proof, see Section 1) might lead one to think that the problem lies in the fact that, as soon as groups are described by algorithms, some version of Rice's theorem, or some reduction of the halting problem, will be found, that prevents one from deducing anything from those descriptions.

The second result of this paper proves the contrary: our most important theorem gives a characterization of finitely presented groups in terms of solvability of two decision problems. For this purpose, we define what we call a marked quotient algorithm of a group:

Definition 2. For a group G generated by a finite set S, we call a *marked quotient algorithm* for G and S an algorithm that takes as input a r.e. algorithm $\mathcal{A}_{re}^{H;S'}$ of another group H over a generating family S', together with a function f that is onto from S to S', and stops if and only if f can be extended to a group homomorphism from G onto H.

We denote by $\mathcal{A}_{Q}^{G,S}$ a marked quotient algorithm for G over S, and omit writing G and S when possible.

Note that there already exist concepts known as "quotient algorithms", those are algorithms that are able to decide whether or not a given group has a quotient in a certain

family of groups. For instance, it was proven in [BW15] that no quotient algorithm exists for the class of all finite groups, but in [BELS19] are given several infinite families of finite simple groups for which quotient algorithms do exist.

We can now state our main theorem:

Theorem 3. *Any group that admits both a r.e. algorithm and a marked quotient algorithm is finitely presented.*

What's more, from any finite presentation of a group G, a pair of algorithms $(\mathcal{A}^G_{re}, \mathcal{A}^G_Q)$ for this group can be effectively computed, and, conversely, from any pair $(\mathcal{A}^G_{re}, \mathcal{A}^G_Q)$, a finite presentation of the group they define can be effectively retrieved.

This theorem is related to Higman's embedding theorem (it is much easier to prove) in the following way: Higman's Theorem shows that, when studying topology, the notion that comes from computability theory of a recursively presented group arises naturally -and even that recursively presented groups could be defined without needing the notions of Turing machines and of computable functions, only in terms of finitely generated subgroups of fundamental groups of closed manifolds. Our theorem does the opposite: it shows that, building a theory of algorithmic descriptions of groups, one could still arrive at the notion of "finitely presented group", solely in terms of algorithms, without even defining group presentations.

We can now state the purpose of this article. We propose to systematically study decision problems for groups described by algorithms, and that the concept of a "global algorithmic problem", which was intended by Dehn to be:

- the study of the properties that can be inferred on a group, given a finite presentation of it, assuming this given group belongs to some specified class of groups;

become:

- the study of the properties that can be inferred on a group, *given any algorithmic description of it*, assuming this given group belongs to some specified class of groups.

We now precise what we mean by an algorithmic description of a group.

The algorithmic descriptions that are easiest to think of arise from the study of local decision problems for groups: if a group G has solvable word problem, an algorithm $\mathcal{A}_{WP}$ that encodes a solution to the word problem with respect to a generating family of G defines G uniquely (in fact a marking of G, we come back to this fact later on). Similarly, the r.e. algorithm of a group defines it uniquely, and so does an algorithm associated to a solution to the conjugacy problem, etc.

This defines a wide range of algorithmic descriptions of groups, that all have a common problem: when trying to solve decision problems for groups based on these descriptions, the balance between the unsolvable and the solvable is overly in favor of the unsolvable. Witness of this are: the already mentioned Proposition 1, and the fact that the isomorphism problem is unsolvable for cyclic groups described by word problem algorithms (which also solve the conjugacy problem). This was proven by Jody Lockhart in [Loc81], we recall it in Proposition 17.

What allows us to get out of this predicament is the marked quotient algorithm introduced for Theorem 3. The marked quotient algorithm of a group G is not attached to a local problem about G, but to the *global* decision problem associated with the group property "being a quotient of G", this problem being asked for groups described by r.e. algorithms.

It is this remark, that the solutions of global algorithmic problems can sometimes be taken as descriptions of groups, that will allow us to find a wide variety of algorithmic group descriptions, rich enough to recover all that can be done thanks to finite presentations, and to open up the possibility of finding descriptions that cannot be expressed purely in terms of presentations.

We can describe a hierarchy of decision problems for groups, associated to a hierarchy of algorithmic descriptions of groups, as follows.

The local decision problems are considered to be the problems of order 0. An algorithm that encodes the solution of a local problem in a group is a description of this group, which is also of order 0.

We can now consider order 1 decision problems, which are global decision problems asked about group descriptions of order 0, and similarly order 1 descriptions of groups, which are the descriptions of groups that arise as solutions of order 1 problems.

We can then define order 2 decision problems, and so on.

As was shown in Theorem 3, finite presentations of groups are descriptions of order 1. And thus the isomorphism problem for finitely presented groups is a decision problem of order 2. An algorithm that takes as input finite presentations of groups and stops only on those that define a certain group G constitutes a description of order 2 of G. Etc.

A very interesting consequence of the duality we introduce between decision problems and descriptions of groups is that, because, as we already mentioned, decision problems for groups are meant to be studied in their rapport with various classes of groups, algorithmic descriptions of groups attached to restricted classes of groups will naturally appear, as the solutions to these restricted problems.

And here we touch on what is perhaps the point where this article lacks the most: we are unable to produce a class of groups in which the *algorithmic* notion of "finite presentation with respect to this class", that is to say the description of a group G that is constituted of a solution to the problem "being a quotient of G", but *asked only for groups in this class*, together with a r.e. algorithm, does not in fact correspond to an already existing concept, already associated to finite presentations -for instance, finite presentations in varieties of groups (see Theorem 43).

It seems to us that the possibility of defining an actual algorithmic generalization of finite presentations would justify our going out of the usual frame of "the study of finitely presented groups", and it hinges on the finding of such a class of groups.

We call *relative marked quotient algorithms* the algorithms associated to the problem of recognizing the marked quotients of a group amongst groups that belong to a specified set, and discuss them in more details in Section 4.

The other important issue we encounter is that we are not able to give a formal definition for what is "an algorithmic description of a group", and the hierarchy that was described above is an informal object.

However, even if it were impossible to resolve, this lack of precise definitions is not a fatal issue: our definitions are only as "informal" as those of Miller, whose definition of local problems involved deciding whether elements of a group had "certain properties or relationships". If a definition of what is an "algorithmic description of a group" cannot be found, we are content with studying explicit examples of such descriptions, which abound, and to propose some new ones.

To propose new descriptions of groups, which would hopefully provide interesting theories of decision problems for groups, we rely on Theorem 3. Recall that it states that a finite presentation of a group can be seen as being a pair:

$$(\mathcal{A}_Q^G, \mathcal{A}_{re}^G)$$

where $\mathcal{A}_{re}^G$ is a r.e. algorithm for G, and $\mathcal{A}_Q^G$ is what we have called a marked quotient algorithm. There are two main directions that this theorem indicates to build interesting algorithmic descriptions of groups. The first one consists in weakening the marked quotient algorithm $\mathcal{A}_Q^G$, and replacing it with a marked quotient algorithm *relative to a class* $\mathcal{C}$. As we have already said, this is discussed in more details in Section 4, and we do not know yet whether this will prove fruitful.

The second path designated by Theorem 3 consists in *strengthening* the right hand side algorithm that constitutes the finite presentation, the r.e. algorithm. This is natural, because this algorithm provides very little information. For instance, semi-computable reals, which, as we already explained, are similar to r.p. groups, appear in other domains of

mathematics mostly as encoding unsolvable problems (the halting problem, the computation of the Kolmogorov complexity of strings), and are not expected to be the appropriate basic objects for the defining of a theory of computable functions of the reals.

In that respect, it could be argued that the most natural algorithmic description of a group is not a single finite presentation for this group, but a finite presentation together with a word problem algorithm. In turn, could be studied algorithmic problems asked for groups described by finite presentations and conjugacy problem algorithms, and order problem algorithms, etc.

This approach is very promising: in fact, it was already successfully applied. In [GW09], Daniel Groves and Henry Wilton proved that, given a finite presentation of a group, together with a solution to the word problem for this group, it is possible to decide whether or not it is a limit group (see [GW09] for a definition of limit groups), and whether or not it is a free group. In [GMW12], those two authors and Jason Fox Manning have showed that from the same description of a group, it is possible to tell whether or not it is the fundamental group of a geometric three manifold (basing their work on [Man02]). Note that the result of [GW09] that concerns free groups was also obtained in [Tou18], where is studied the computability of Grushko decompositions for groups described by finite presentations together with word problem algorithms. We summarize those results in the following theorem:

Theorem 4 ([GW09],[GMW12],[Tou18])**.** *Given a group, described by a finite presentation together with a word problem algorithm, it is possible to decide whether or not this group is free, is a limit group, or is the fundamental group of a geometric three manifold.*

It is not only those results that are interesting, but also the methods of proofs involved. For instance, consider only the statement of this theorem that concerns free groups, it answers what may be the most basic problem about groups described by finite presentations together with word problem algorithms: is it possible to recognize free groups from this description?

The answer to this question given in [GW09] relies on the study of limit groups, and in fact on many elaborate results about limit groups (finite presentability and coherence of limit groups, Makanin's Theorem which establishes the decidability of the universal theory of free groups, etc).

We think it is fair to argue that the very interesting methods involved in the proof of Theorem 4 justify, *a posteriori*, that the question which is answered in this theorem was an interesting one.

There are some differences between our formalism and the one that is used in [GMW12]. It is however not difficult to see that the proofs of [GMW12] do imply Theorem 4. In Section 2, we discuss in more details the formalism of [GMW12] and explain why we prefer ours, using our analogy with computable analysis: the computability notion of [GMW12] resembles that of Banach-Mazur computability, which was supplanted, in computable analysis, by other notions of computability.

The fact that, for the description "finite presentation-word problem algorithm", there exist groups that are recognizable, and some that are not (by [Mil72], this is discussed in Section 3), shows that the study of group recognition is more interesting when based on this description than on the one associated to finite presentations, since it is well known that the Adian-Rabin theorem implies that the problem "does this presentation define G?" is always unsolvable in the class of all finitely presented groups, even when G is the trivial group.

We talk more about group recognition in Section 3.

Overall, it seems that the study of decision problems for groups given by descriptions that are stronger than finite presentations should prove fruitful.

We now formulate our last remark about the question of what a theory of decision problems for groups, separated from topology, could look like: it seems to us that in such a theory, finitely presented groups still have to play a central role, not because they are the objects that such a theory aims at studying, but because they serve as the model of a

class of groups, associated to a group description, in which a satisfying theory of decision problems can be built.

Decision problems for marked groups. Another point we want to make in this article is the following: the appropriate category to study decision problems for groups is the category of marked groups.

Our first argument in this direction is given to us by Dehn's definition of the isomorphism problem ([Deh87]):

> "The Isomorphism Problem: Given two groups, one is to decide whether they are isomorphic or not (and further, whether a given correspondence between the generators of one group and elements of the other group is an isomorphism or not)."

Thus Dehn asked for both a solution to the usual isomorphism problem, *and* a solution to the "marked isomorphism problem". This last aspect seems to have been all but forgotten later on.

A *k-marked group* is a finitely generated group together with a k-tuple of elements that generate it. A morphism between k-marked groups is a group morphism that sends the generating tuple of the first group to the generating tuple of the second group. Such a morphism is an isomorphism if the group morphism is a group isomorphism, and of course marked groups are considered up to isomorphism. Notice that the relation $G \geq H$, defined for marked groups G and H by "there is a morphism of marked groups from G to H", defines an order on the set of marked groups, it is easy to see that it admits both meet and join operations. The set of k-marked groups equipped with this order is thus a lattice. For more about marked groups, see [CG05].

It is in fact convenient, when studying k-marked groups, to fix a free group F of rank k, together with a basis S. A k-marking of a group G can then be seen as an epimorphism $\varphi : F \to G$, the image of S by φ defines a marking with respect to the previous definition. Two k-marked groups are then isomorphic if they are defined by morphisms with identical kernels: the isomorphism classes of k-marked groups are in bijection with the normal subgroups of a rank k free group. This is nicely related to algorithmic descriptions of groups: the basis S of the free group F can be seen as a set of tape symbols which will be common to all algorithms that describe groups.

We call a group an *abstract group* when we want to emphasize the fact that it is not a marked group.

Note that all the algorithmic descriptions of groups we have encountered so far provide descriptions of marked groups[1]. Finite and recursive presentations, word problem algorithms, conjugacy problem algorithms, etc, all define marked groups. The marked quotient algorithm which we introduced is also attached to a marked group, even more: it provides information about the location of a given marked group in the lattice of marked groups.

It does not seem natural, working with descriptions that define marked groups, to always discard the additional information provided by the "markings", and only consider problems about the underlying abstract groups.

More importantly, as long as no groups descriptions are used, that are attached to abstract groups and do not define marked groups, the study of decision problems for marked groups strictly contains the study of abstract decision problems. That is because a property of abstract groups is simply a property of marked groups that is *saturated* ([CG05]), that is to say which is invariant under abstract group isomorphism. Thus the study of decision problems for marked groups does not remove anything to the study of decision problems for abstract groups, and on the contrary, one can gain a better

[1]There is actually one exception: the "order 2" description that we introduced above, which consists in an algorithm that recognizes the finite presentations of a given group. We don't know wether such a description can be used in practice, and provide interesting results. Of course, the study of descriptions of marked groups will not prevent one from studying abstract groups descriptions when they arise.

understanding of decision problems for abstract groups by having studied more generally decision problems for marked groups.

For instance it can be very helpful to be able to tell when a problem is undecidable, both for marked and abstract groups, or when it is decidable for marked groups, but undecidable for abstract groups, and that the undecidability arises precisely during the transition from the category of marked groups to the category of groups.

The following is an example of this kind.

The Adian-Rabin theorem implies unsolvability of the isomorphism problem for abstract groups, but it is in fact usually proved by relying on a single marked groups: a sequence $(\pi_n)_{n\in\mathbb{N}}$ of finite presentations is constructed, such that the set of indices n for which π_n defines the same *marked group* as π_0 is recursively enumerable but not recursive. This can be done because the marked isomorphism problem is unsolvable for finitely presented groups. On the contrary, in Miller's proof from [Mil72] (Theorem 26, Chapter IV) of the unsolvability of the isomorphism problem for finitely presented residually finite groups, a fundamental difference appears, because the marked isomorphism problem is solvable for such groups (by [McK43], see Section 3). In [Mil72] is constructed a sequence $(\pi_n)_{n\in\mathbb{N}}$ of finite presentations of residually finite groups such that the set of indices n for which π_n defines some fixed *abstract group* G is recursively enumerable but not recursive, to achieve this, the presentations that do define G must be attached to infinitely many generating families of G, families whose average word length, with respect to some fixed generating family of G, grows faster than any recursive function.

We give one last example which shows, in our opinion, that the study of decision problems for marked groups was actually never very far. It again relies on the Adian-Rabin theorem.

When stating this theorem, many authors, to define a *Markov property*, state something along the lines of: "a Markov property is a group property which *is invariant under group isomorphism*, and such that...". See Rabin himself in [Rab58], and for instance [Rot99], [Bau93], [BK89] and [Mil92].

For instance, in Miller's article which we have already cited ([Mil92]), one can find the following paragraph:

"Consider the problem of recognizing whether a finitely presented group has a certain property of interest. For example, can one determine from a presentation whether a group is finite? or abelian? It is natural to require that the property to be recognized is abstract in the sense that whether a group G enjoys the property is independent of the presentation of G."

Now notice that it does not make sense to require that *a group property* be invariant under isomorphism, by *definition* a group property is invariant under group isomorphism. And thus to make sense of what Miller calls a "property of finitely presented groups", we might have to consider that it is actually *a property of finite presentations*. But it seems clear that properties of finite presentations that are completely detached from the groups they define are not really worth mentioning. Properties of marked groups are precisely the right intermediate between properties of finite presentations and properties of groups which allow to make sense of the requirement that a Markov property be a property of abstract groups: a property of abstract groups, as opposed to a property of marked groups.

We point out the following consequence of our introducing decision problems for marked groups: while the Adian-Rabin Theorem completely solves the problem of abstract group recognition for finitely presented groups (in the class of all finitely presented groups), and while the proof of this theorem can be adapted to obtain results about the recognition of marked groups, there is a gap in the results thus obtained about marked groups, which seemingly would require different techniques to be solved. We talk about this in Section 3.

By now, most of the content of this article was already presented. We end this introduction by giving summaries of the contents of our different sections.

In Section 1, we prove Proposition 1 as well as Theorem 3. We also propose three more undecidability results that are inspired by Proposition 1: one for groups described by what is known as co-recursive presentations, one for groups described by marked quotient algorithms, and finally, the already mentioned result of Lockhart in [Loc81] about groups given by word problem algorithms.

In Section 2, we provide some additional discussion about possible critics to our proposed approach to decision problems for groups: the opportunity of considering decision problems that take "well chosen" recursive presentations as input, the defining of problems that are not defined on recursively enumerable sets of groups. We also discuss the possibility of establishing equivalents of the Kreisel-Lacombe-Schoenfield-Ceitin Theorem for algorithms that describe groups, and finally we talk about the notion of "recursive modulo the word problem" that comes from [GMW12].

In Section 3, we study the consequences of our proposed approach to decision problems for groups, in the particular case of group recognizability: the problem of recognizing a fixed group. This is a natural step towards understanding the isomorphism problem. We explain there the advantage of finite presentations over other groups descriptions in terms of "uniform semi-recognizability".

We propose several problems there as well.

Our final section, Section 4, is dedicated to establishing some basic results about relative marked quotient algorithms. We remark that Theorem 3 can be applied in group varieties, define what we call "elementary marked quotients algorithms", (those are the relative marked quotients algorithms that could be defined thanks to finite presentations), and give a simple example of a non-elementary marked quotient algorithm: the finite quotient algorithm of the lamplighter group.

1. Recursive and finite presentations

1.1. **Solving decision problems from recursive presentations.** As opposed to when groups are described by finite presentations, in which case although the isomorphism problem is unsolvable in general, it can become solvable if we only consider groups that satisfy some constraints (either algebraic, or geometric, etc), if we wish to solve the isomorphism problem in some classes of groups given by arbitrary recursive presentations, the problem can almost never be solvable, even with dire restrictions on the considered groups.

Proposition 5. *There exists a recursively enumerable sequence of recursive presentations, defined on one generator, that define either the trivial group, or the group of order two, and such that the set of indices n for which π_n defines the trivial group is recursively enumerable but not recursive.*

Proof. Consider an effective enumeration of all Turing machines, M_0, M_1, M_2, M_3,... We define for each natural number n a presentation π_n. The presentation π_n has a single generator a, and starts with a single relation a^2. An effective enumeration of the relations of π_n is defined thanks to a run of the machine M_n. While this run lasts, no relators are added to π_n. If it stops, the relation $a = 1$ is added. Thus if M_n stops, π_n is of the form:

$$\pi_n = \langle a | a^2, a \rangle$$

If M_n runs forever, π_n is the finite presentation:

$$\pi_n = \langle a | a^2 \rangle$$

(Note that thanks to this definition, the relations of the presentations π_n form a recursively enumerable set, and this uniformly in n, but not a uniformly recursive set, as we cannot know beforehand whether a is a relation. Replacing a by a^{2k+1} in π_n, where k is the number of steps needed for the machine M_n to stop, makes these uniformly recursive presentations.)

This is clearly a recursively enumerable set of presentations: they were defined thanks to a procedure that can be carried out effectively.

And of course, no algorithm can decide, given some n, whether π_n defines the trivial group, since it defines the trivial group if and only if the n-th Turing machine stops. □

Remark that the presentations written above are both finite... However, they are not given by the finite number of symbols that constitute the presentations, but by Turing machines, which, when run, will output the finitely many relations of the presentations. Thus there is no contradiction between our proof and the fact that the isomorphism problem is solvable for finite groups given by finite presentations. We discuss this in more details in Section 2, where we try to justify that this proposition was not obtained thanks to our using ambiguous definitions, but that on the contrary it indeed shows that the use of recursive presentations is impractical.

The previous proposition can be applied to both the study of abstract and of marked groups, because all the descriptions used define the same markings of the considered groups.

It is easy to see that the construction that appears in the proof of Proposition 5 can be applied, not only to the trivial group and the order two group, but to *any pair of recursively presented groups where one is a quotient of the other*. This yields the following proposition, which is almost the same as Proposition 1, which appeared in the introduction, but we precise it so that it be applicable to the study of marked groups as well as that of abstract groups.

Proposition 6. *Let G and H be finitely generated, recursively presented groups, and suppose that H is a strict quotient of G. Fix a marking of G, and consider the induced marking on H. Then, the problem of deciding whether a given pair of recursive presentations, that define either the chosen marking of G, or that of H, define the same group, is unsolvable.*

Proof. It is easy to adapt the proof of Proposition 5: to define π_n, enumerate the relations of G as long as the Turing machine M_n runs, if it stops, add the relations of H. □

It is not the case that any two groups are undistinguishable with respect to the recursive presentation description. It is easy to see that from a recursive presentation over one generator a, that defines either the order two group, or the order three groups, one can determine which one it is, because enumerating the consequences of the relations, one of a^3 or a^2 will eventually appear. More generally, it was shown by Kuznetsov ([Kuz58]) that the word problem is uniformly solvable for simple groups described by recursive presentations. And it is easy to see that the isomorphism problem is solvable for any finite set of finitely presented simple groups *described by recursive presentations,* but of course the class of simple groups does not contain a group and a strict quotient of it.

As a direct consequence of this last proposition, we get the following corollary, which may be seen as an equivalent of the well known Rice theorem, from computability theory:

Corollary 7. *No non-trivial group property can be determined for groups given by recursive presentations.*

Proof. It suffices to notice that for any non-trivial group property, (that is a property some group enjoy, while others lack), there must exist a pair of groups G and H, such that H is a quotient of G, and one satisfies the property, while the other does not. □

Corollary 8. *Any abstract or marked group property which is partially decidable, in a class $\mathcal{C}$ of groups, for groups given by recursive presentations, is quotient-stable.*

Here, by partially decidable, we mean that there is an algorithm that stops exactly on recursive presentations of groups with that property.

Proof. This follows from the fact that, for the sequence of presentations constructed in the proof of Proposition 6, there is an algorithm that stops only on recursive presentations of the quotient group H. Thus there cannot be an algorithm that stops only on the recursive presentations of the group G. □

Note that group properties which are partially decidable from recursive presentations do exist: being trivial, being finite, or, a non trivial example, having Kazhdan's property (T) ([Oza14]).

1.2. **The marked quotient algorithm and finitely presented groups.** Recall that the notation $\mathcal{A}_Q^{G;S}$ designates a marked quotient algorithm for the group G over the generating set S, as defined in the introduction.

Although we stated in Definition 2 that such an algorithm takes as input both a r.e. algorithm $\mathcal{A}_Q^{H;S'}$ for a groups H, and a function f from the generating family S of G onto that of H, we will in fact often suppose that both groups G and H are generated by S, this allows us to omit the function f and to consider that the input of a marked quotient algorithm is a single r.e. algorithm.

Notice first that any finitely presented group admits a marked quotient algorithm. Indeed, to know whether a group H is a marked quotient of a finitely presented group G, one only needs to check whether the finitely many relations of G hold in H. Given a finite set R of relations that define G over the generating set S, and the r.e. algorithm $A_{re}^{H;S}$, the marked quotient algorithm of G computes the boolean operation:

$$\underset{r\in R}{\&} A_{re}^{H;S}(r)$$

Notice however that finitely presented groups are not the only groups that admit marked quotient algorithms: a non-recursively presented simple group also has one -its only recursively presented quotient is the trivial group. However, Theorem 3 states that they are the only ones amongst recursively presented groups.

Theorem (Theorem 3)**.** *Any group that admits both a r.e. algorithm and a marked quotient algorithm is finitely presented.*

What's more, from any finite presentation of a marked group G, a pair of algorithms $(\mathcal{A}_{re}^G, \mathcal{A}_Q^G)$ for it can be effectively computed, and, conversely, from any pair $(\mathcal{A}_{re}^G, \mathcal{A}_Q^G)$, a finite presentation of the marked group they define can be effectively retrieved.

Proof. The fact that a pair $(\mathcal{A}_{re}^G, \mathcal{A}_Q^G)$ can be effectively obtained from a finite presentation of G was already explained: $\mathcal{A}_{re}^G$ lists the consequences of the relations of G, and $\mathcal{A}_Q^G$ tests whether the relations of G hold in an input group.

Consider a marked group G which admits a description $(\mathcal{A}_{re}^G, \mathcal{A}_Q^G)$. Consider an enumeration of all Turing Machines M_0, M_1... For each natural number n, we define an algorithm $\mathcal{A}_{re}^n$. It is defined as enumerating the relations of a group G_n, rather than as recognizing relations. $\mathcal{A}_{re}^n$ does two actions in parallel: it simulates the machine M_n, all the while listing relations of G thanks to $\mathcal{A}_{re}^G$. If M_n stops, $\mathcal{A}_{re}^n$ stops using $\mathcal{A}_{re}^G$, and proceeds to listing all the consequences of the finitely many relations it has already obtained. On the contrary, if M_n does not stop, $\mathcal{A}_{re}^n$ outputs exactly the same list as $\mathcal{A}_{re}^G$.

Notice now that $\mathcal{A}_Q^G$ will accept every algorithm $\mathcal{A}_{re}^n$ corresponding to a non-halting Turing Machine, since it must accept the identity $\mathrm{id} : G \to G$.

But of course, it cannot accept only those, since it would otherwise allow one to solve the halting problem. Thus it will accept some $\mathcal{A}_{re}^n$ for a Machine M_n that stops in a finite amount of time. This implies that the relations of G are in fact all consequences of the first few relations that $\mathcal{A}_{re}^G$ is able to produce during a run of this machine M_n, and so that G is finitely presented.

Because the indices of halting Turing Machines can be listed, as well as the indices that define quotients of G, some integer n_0 that satisfies both those conditions can be effectively found, and from it, also the finitely many relations that define G over S.

Now notice that the construction described above is uniform, and allows one to obtain a procedure that takes as input a pair $(\mathcal{A}_{re}^G, \mathcal{A}_Q^G)$ and outputs a finite presentation for G. □

This theorem precisely means that every global algorithmic problem that was solved for finitely presented groups could have been solved with the two algorithms described in

the theorem as initial data, and thus no intrinsic unsolvability lies in the use of algorithms to describe groups.

We now prove the very important (and well known) fact, that from any finite presentation of a group, any other presentation of the same group can be recognized, as is usually done by listing Tietze transformations ([Tie08]), but using the point of view given by Theorem 3.

Proposition 9. *There is an algorithm that takes as input pairs of finite presentations, and stops exactly on pairs of presentations that define the same (abstract) group.*

Proof. From the point of view of algorithmic descriptions, only the Tietze transformations that introduce and delete generators are needed -those that change the marking of the considered group. Indeed, testing whether two finite presentations $(\mathcal{A}^G_{re}, \mathcal{A}^G_Q)$ and $(\mathcal{A}^H_{re}, \mathcal{A}^H_Q)$ over the same generating set S define the same marked group is done directly by the computation of $A^G_Q(A^H_{re})\&A^H_Q(A^G_{re})$.

All generating families of a group can be enumerated using only the r.e. algorithm, this is easy to see: choose arbitrary words in the generators, and then blindly search for an expression of the original generators in terms of products of those random words. This procedure terminates on all generating families of the group. When a finite set of elements is found to be a generating family, it means that one has access to expressions both of the new generators expressed as products of the old ones, and of the old generators expressed as products of the new ones. By the following lemma, Lemma 10, we can obtain a marked quotient algorithm with respect to any generating family we find, and thus recognize any finite presentation of a given group. □

Lemma 10. *A finitely generated group which admits a marked quotient algorithm with respect to a marking admits one for all of its markings. Such algorithms can be obtained one from the other if one has access to expressions that describe the elements of each generating family in terms of products of elements of the other family.*

Proof. Let S and T be two finite generating sets of a group G (not necessarily of the same cardinality). We suppose that we have access only to A^S_Q, the marked quotient algorithm for G with respect to S. Fix for each s in S an expression $s = t_1^{\alpha_1}...t_k^{\alpha_k}$, with $\alpha_i \in \{-1, 1\}$ and $t_i \in T$, that gives s as a product of elements of T and of their inverses, and for each t in T an expression $t = s_1^{\beta_1} s_2^{\beta_2}...s_k^{\beta_k}$ that describes t in terms of the generators of S and their inverses.

Consider a group H generated by the same family T, given by its r.e. algorithm $A^{H;T}_{re}$, we want to determine whether (H, T) defines a marked quotient of (G, T).

Notice that if the identity on T defines a morphism ϕ of G onto H, the family S', defined, in H, by the same formulas as S in G, i.e. $s' = t_1^{\alpha_1}...t_k^{\alpha_k}$, should be the image of the family S by ϕ, and thus should be a generating family of H. We can therefore, using the algorithm $A^{H;T}_{re}$, look for an expression of the elements of T in terms of the elements of S' in H. If such an expression does not exist, our procedure will not stop, but H is not a quotient of G, thus this result is coherent. Otherwise we can use the formulas just found to obtain the r.e. algorithm for H with respect to S', $A^{H;S'}_{re}$. From it, we can ask whether the natural bijection $S \to S'$ defines a morphism, applying $A^S_Q(A^{H;S'}_{re})$. If this procedure does not end, H was not a quotient of G. If it terminates, in which case $S \to S'$ does define a morphism ψ, we still have to check whether ψ defines the identity on T. This is done using the expressions of the form $t = s_1^{\beta_1} s_2^{\beta_2}...s_k^{\beta_k}$, that define, in G, the elements of T in term of those of S, the corresponding expressions in H of elements of T in terms of elements of S', and the algorithm $A^{H;S'}_{re}$. One can check that the given conditions are necessary and sufficient for the identity of T to extend as a homomorphism. □

Other examples in the same vein as this one can be found, where we prove results that are usually explained thanks to the manipulation of finite presentations, using only algorithms -without using Theorem 3. Easy examples are: computing the presentation of a free product, or the abelianization of a group. On the contrary we do not know how

to obtain an enumeration of all finitely presented groups, described by pairs (A_{re}, A_Q), as is obtained by listing all possible finite presentations, without precisely listing those presentations and applying Theorem 3.

The manipulation of the algorithms that constitute a finite presentation, while possible, is often less convenient than the manipulation of actual presentations. For instance, we do not believe that the algorithmic description of finite presentations can ever be used to prove that a group is or is not finitely presented.

Remark 11. The marked quotient algorithm, as defined, halts only on accepted inputs. Are there marked groups that could have a always halting marked quotient algorithm, which would answer "yes" or "no" depending on whether the input is a quotient of the considered group? Free groups, of course, can: the marked quotient algorithm for a free group over a basis accepts all inputs. This is, however, the only possible example, because if a non-free finitely presented group G admitted a always halting marked quotient algorithm with respect to a generating family S, the isomorphism problem would be solvable for the pair constituted of G and of the free group on S, *when described by recursive presentations*. By Proposition 6, this is impossible.

On the other hand, it is easy to remark that, given a finitely presented group G, one can obtain a modified marked quotient algorithm for G that always halts, by accepting groups described, not by their r.e. algorithm, but by their *word problem algorithm*. Instead of having to wait maybe infinitely long before knowing whether the finitely many relations of G hold in a group H, an answer will be found in a finite amount of time. This new type of algorithm is what we call a *marked quotient algorithm with Word Problem as Input*, which we will abbreviate as WPI marked quotient algorithm. Note, however, that such quotient algorithms do not characterize finitely presented groups amongst r.p. groups anymore. Indeed, Miller constructed (see [Mil92]) a finitely presented group which has unsolvable word problem, such that no non-trivial quotient of it has solvable word problem. One can obtain such a group that is r.p., but not finitely presented, using exactly the same construction... except for Higman's embedding theorem. This group admits a marked quotient algorithm that takes as input word problem algorithms and answers whether or not the corresponding group is a quotient: this algorithm always answers no, except on the trivial group.

All this sums up why Theorem 3 uses the marked quotient algorithm which, by nature, does not always stop.

Problem 12. Characterize the groups with solvable word problem that admit an WPI marked quotient algorithm: are these only the finitely presented groups with solvable word problem?

More variations of the quotient algorithm are studied in Section 4.

1.3. **Three more unsolvability results.**

1.3.1. *From the co-r.e. algorithm.* The interest in finitely presented groups, together with Higman's Theorem, makes being recursively presented a much more natural notion to introduce than the notion known as being co-recursively presented ([Man82]), which, contrary to what the names indicates, is not defined thanks to a presentation: a marked group is co-r.p. if there is an algorithm that recognizes the words in the generators of this group that define non-identity elements, and because this property of marked groups is saturated, it defines a group property. We call such algorithms co-r.e. algorithms, for *co-recursive enumeration algorithm,* and denote by $\mathcal{A}^{G,S}_{co-re}$ a co-r.e. algorithm for the group G generated by a family S.

From the algorithmic point of view, the notions of recursive and co-recursive presentations are more or less symmetrical. For instance, using our analogy with computable analysis, note that if r.p. groups correspond to lower semi-computable reals, co-r.p. groups will then correspond to upper semi-computable reals. (There is however nothing that would correspond to the bijection $x \mapsto -x$ that sends lower to upper semi-computable reals.)

This remark calls for the following proposition:

Proposition 13. *Let G and H be finitely generated, co-recursively presented groups, and suppose that H is a strict quotient of G. Fix a marking of G, and consider the induced marking on H. Then, the problem of deciding whether a given pair of co-r.e. algorithm, that define either the chosen marking of G, or that of H, define the same group, is unsolvable.*

Proof. As for Proposition 6, we prove this result for the trivial group and the order two group, and note that the same construction can be applied to an arbitrary pair.

To the n-th Turing machine M_n, associate the pair composed of one generator symbol a, and of an algorithm $\mathcal{A}^n_{co-re}$ defined as follow: to determine whether a word a^k, with k an integer, is a non-identity element, $\mathcal{A}^n_{co-re}$ starts by simulating the machine M_n. If it stops, $\mathcal{A}^n_{co-re}$ determines whether k is even or odd, and if k is odd, it indicates that a^k corresponds to a non-identity element.

If M_n does not stop, the algorithm $\mathcal{A}^n_{co-re}$ never recognizes any element -it is just an algorithm that runs forever, it thus defines the trivial group. On the other hand, if M_n stops, $\mathcal{A}^n_{co-re}$ recognizes exactly the non-identity elements of the group of order two. □

As before, this result implies a Rice Theorem for groups given by algorithms that recognize non-trivial elements:

Corollary 14. *No non-trivial group property can be determined for groups given by their co-r.e. algorithm.*

Corollary 15. *If P is a marked or abstract group property which is partially decidable for groups given by their co-r.e. algorithm, in a class $\mathcal{C}$ of groups, any extension of a group with P also has P.*

An example of such a property is: being non-trivial.

1.3.2. *From the marked quotient algorithm.* We will now prove another unsolvability result that is intended to show that, in Theorem 3, it is necessary to include both the marked quotient algorithm and the r.e. algorithm. Notice that the marked quotient algorithm of a finitely presented group defines it uniquely: if G and H have the same marked quotient algorithm A_Q and are r.p., it must be that $A_Q(A^G_{re})$ and $A_Q(A^H_{re})$ both are accepted, thus G and H are isomorphic (even as marked groups). However, this algorithm gives very little information:

Proposition 16. *No algorithm can solve the isomorphism problem for the trivial group and the order two group when they are described by their marked quotient algorithms.*

Proof. Again, to the n-th Turing machine M_n, associate the pair composed of one generator symbol a, and of an algorithm $\mathcal{A}^n_Q$ defined as follow: to determine whether an algorithm $\mathcal{A}^H_{re}$ defines a quotient of our group, try to determine if H is trivial, by waiting to see whether $\mathcal{A}^H_{re}$ accepts a. Of course, if at some point it is proven that H is the trivial group, stop and accept. All the while, start a run of the machine M_n. If it stops, accept also the order two group as a quotient: try to see whether $\mathcal{A}^H_{re}$ recognizes a^2 as a relation, and accept if it does.

It is clear that if M_n stops, this defines the order two group, while it defines the trivial group otherwise. □

Notice how we could not have made simultaneously this construction and that of Proposition 5: here the non-stoping Turing machines *must* correspond to the trivial group, whereas in Proposition 5, they had to correspond to the order two group. This proposition could be extended to any pair of groups as before.

1.3.3. *From the word problem algorithm .* The last result on this list concerns the word problem. We include it here because it is relevant in showing that the solutions to local algorithmic problems provide weak algorithmic descriptions.

Proposition 17. *No algorithm can solve the isomorphism problem for cyclic groups when they are described by their solution to the word-problem.*

Proof. Consider an enumeration of all Turing Machines M_0, M_1,... For each natural number, define an algorithm A^n_{WP} that works with a single generating symbol a as follows. Consider as input an element a^k, $k \in \mathbb{Z}$. If $k = 0$, then of course a^k is accepted as trivial. Otherwise A^n_{WP} simulates a $|k|$ steps run of M_n. If it does not stop during this run, than A^n_{WP} answers that a^k is a non-identity element. If on the contrary M_n stops in less than $|k|$ steps, say in p steps, we decide that A^n_{WP} should be the word problem algorithm for the group $\mathbb{Z}/p\mathbb{Z}$, and thus that it should answer that a^k is the identity if and only if p divides k.

It is easy to see that this definition is coherent, that is to say that A^n_{WP} will always answer as the word problem algorithm of the same group, whatever the input.

And A^n_{WP} defines $\mathbb{Z}$ if and only if M_n does not stop. □

The proof above is very similar to a proof of Lockhart in [Loc81]. However, in Lockhart's article, the statement which is proved, thanks to this proof, is the following: "There is a r.e. class of recursive presentations with uniformly solvable word problem for which the properties of freeness and finiteness are unrecognizable". We will discuss the ambiguity of this statement, which is similar to that of Proposition 5, in SubSection 2.1.

In an upcoming article ([Rau21b]), we will study in more details decision problems for groups given by word problem algorithms. In particular, we will show that the proof given above can be interpreted as an application of a result of computable analysis due to Markov Jr. ([Mar54]), which states that computable functions cannot have effective discontinuities, in the topological space of marked groups.

2. Additional discussion

The use of algorithms to describe countable groups is just one amongst many means to obtain finite definitions of potentially infinite groups. We can quote here several approaches to describing groups that are commonly used:

- Finite presentations;
- Finite presentations in group varieties ([KS95]);
- Automaton to describe automatic groups ([EPC+92]);
- Finite sets of axioms to describe Finitely Axiomatizable groups ([Nie07]);
- Finitely generated linear groups described by generating matrices ([DF19]);
- L-presentations ([Bar03]);
- Etc.

Note that algorithmic descriptions of groups were already studied in [BCR19], for groups that are not finitely generated. The results of this article justify our focus on finitely generated groups, by showing that almost no global problem is decidable for infinitely generated groups described by algorithms. Note that, in [BCR19], some interesting problems are still raised in the setting of infinitely generated groups, by going past the problem of solvability of different decision problems, and asking more precisely for their location in the Kleene–Mostowski hierarchy.

In this section, we give some precisions about the point of view taken in the present article, and answer possible critics to it.

2.1. **Recursive presentations.**

2.1.1. *Ambiguity of the concept of recursive presentations.* We stated in the introduction that we consider that Proposition 1 prevents one from basing a theory of decision problems on recursive presentations, and we in fact choose to talk about r.e. algorithms rather than of recursive presentations, even though recursively presented groups are exactly those that admit a r.e. algorithm.

We now explain in more details those statements.

The first important point is the ambiguity that lies in the term "recursive presentation"[2], which appears clearly in the proof of Proposition 5, in which we prove that the trivial group and the order two group cannot be distinguished when described by recursive presentations. This proof relies on only two actual presentations, which are $\langle a|a^2, a\rangle$ and $\langle a|a^2\rangle$.

The ambiguity lies in the following: by definition, a recursive presentation is a presentation $\pi = \langle S|R\rangle$, where R is a recursively enumerable set of words. On the other hand, when we say that a group is "given by a recursive presentation", as input of an algorithm, we mean to say that is given a pair $(S, \mathcal{A}_{rel})$, where S is a set of generators symbols, and $\mathcal{A}_{rel}$ is an algorithm that enumerates the relations of the underlying presentation.

Thus in order to use a recursive presentation to obtain an algorithmic description of a group, a choice has to be made, we precisely take the step that goes from the abstract "there exists an algorithm that outputs the relations of π" to the statement: "$\mathcal{A}_{rel}$ is an algorithm that outputs the relations of π".

We can see that the proof of Proposition 5 precisely relied on making inefficient choices of algorithms that would output the relations of either of the presentations $\langle a|a^2, a\rangle$ and $\langle a|a^2\rangle$: instead of using the obvious algorithms associated to those presentations ("output a^2, then a, then stop", or "output a^2 and stop"), we gave infinitely many possible algorithms which defined $\langle a|a^2, a\rangle$ and $\langle a|a^2\rangle$, most of which were very inefficient: the algorithms that defined $\langle a|a^2, a\rangle$ produced first the relation $a^2 = 1$, then waited for a possibly very long time to output the relation $a = 1$, while the algorithms for $\langle a|a^2\rangle$ produced right away the relation $a^2 = 1$, but instead of stoping, continued running for ever.

The proof of Proposition 5 might seem to be "artificial", or to rely on an error of definition that could be easily avoided, precisely because it relies on poor choices of algorithms. But remark that those choices can be deemed as "poor" only because there are, in the case of the presentations $\langle a|a^2, a\rangle$ and $\langle a|a^2\rangle$, obvious better choices of algorithms that could be made, because those presentations are finite.

But in general, under the sole assumption that π is a recursive presentation, no canonical choice of an algorithm that outputs its relations can be made, whether it be regarding the order in which the relations should be produced, or regarding the time it takes for the algorithm to produce a new relation, etc. And thus there is no hope of excluding the "bad" algorithms that were used in the proof of Proposition 6 if we are to deal with generic recursively presented groups.

Talking about the r.e. algorithm of a group allows us precisely to render explicit the choice of an algorithm that needs to be made, all the while making the only canonical choice that can be made: that the set of enumerated relations be the maximal one.

2.1.2. *Using restricted sets of recursive presentations algorithms.* After reading the previous paragraph, one might still be tempted to base a theory of decision problems for groups on recursive presentations, making use of the above described choice of an algorithm $\mathcal{A}_{rel}$ that gives the relations of a recursive presentation π: while, in general, no a priori properties can be expected of the algorithm $\mathcal{A}_{rel}$, it might be possible to make different assumptions on $\mathcal{A}_{rel}$ when working in different classes of groups. The most obvious example of this is that of finitely presented groups: in case π is a finite presentation, we can ask of $\mathcal{A}_{rel}$ that it halt after a finite computation. And we could expect that in different classes of recursively presented groups, some assumptions could be made about the algorithms $\mathcal{A}_{rel}$, in accordance to the nature of the groups in questions, and that this would prevent one from obtaining too strong undecidability results, such as those of Proposition 1.

[2]We will not mention the fact that a proper name should be "recursively enumerable presentation". The well known trick that allows to replace a recursively enumerable set of relators by a recursive one is not harmless, for instance it requires the use a different marking of the considered group, or, worse, in [GI09] is defined a Dehn function for infinitely presented groups, which is not a group invariant anymore, and which can be used only in groups whose relations form an actually recursive set, and changing a r.e. set of relations into a recursive set of relations will change the behavior of this function...

We now explain why we do not recommend this approach.

Our argument is based on the following example. We already mentioned (in Proposition 17) the result of Lockhart from [Loc81]: "There is a r.e. class of recursive presentations with uniformly solvable word problem for which the properties of freeness and finiteness are unrecognizable". The groups in questions are finite and infinite cyclic groups.

Remark here that Proposition 6 shows that when a class of recursively presented groups contains a group and a strict quotient of it, then no sequence of "actual" recursive presentation of those groups can have uniformly solvable word problem. This is because the marked isomorphism problem is unsolvable for a pair composed of a group and of a quotient of it, when groups are "given by recursive presentations", while it is solvable when those groups are described by word problem algorithms. And indeed what is actually proven in [Loc81] is not a result that concerns a "r.e. class of recursive presentations", but a r.e. sequence of *algorithms*, the choice of a way of enumerating the relations for the described presentations was explicitly made. This remark shows in particular that the ambiguity that lies in the use of the terms "groups given by recursive presentations" had already appeared in the literature before, and that we did not introduce it in order to obtain Proposition 1.

With this example in mind, consider the following: in any class $\mathcal{C}$ of groups with solvable word problem, if we allow ourselves to make choices about the algorithms that produce recursive presentations, then finding a sequence of algorithms that output recursive presentations which has "uniformly solvable word problem" is a trivial matter, as we can choose algorithms that output all the relations for the groups they define, increasingly with respect to some computable order.

Such algorithms exist because the considered groups have solvable word problems... but they are actually not different at all from word problem algorithms. In this context, the sentence "a set of recursive presentations with uniformly solvable word problem" is not different from the sentence "a set of word problem algorithms", and it is clear that the latter should be preferred.

In general, it seems to us that one obtains more precise and more explicit conditions on algorithmic descriptions of groups by using algorithms that perform advanced tasks, than by considering recursive presentations that are assumed to satisfy additional properties.

2.2. **The Kreisel-Lacombe-Schoenfield-Ceitin Theorem.** Our analogy with computable analysis suggests to us the need to establish results that correspond to the Kreisel-Lacombe-Schoenfield-Ceitin Theorem, which is one of the most important theorems of computable analysis.

The Kreisel-Lacombe-Schoenfield-Ceitin Theorem was first proven by Kreisel, Lacombe and Schoenfield on the Baire space in [GK57], Ceitin later extended their results to effectively Polish spaces in [Cei67]. See for instance [PDD16] for a presentation of those results.

We do not state here this theorem in its most general setting, but only as applied to the set of computable real numbers.

Theorem 18 (Kreisel-Lacombe-Schoenfield-Ceitin Theorem)**.** *One does not change the set of "computable functions defined on the computable reals" if one considers either of the followings definitions:*

- *a function f is computable if there is a Turing machine that takes as input another Turing machine that, on input n, produces a rational approximation of a real x with error at most 2^{-n}, and produces as output a Turing machine which similarly gives approximations of $f(x)$;*
- *a function f is computable if there is a Turing Machine that, given access to an infinite Tape (an oracle), on the n-th cell of which is written an approximation within 2^{-n} of a real x, will write on an output Tape a sequence of approximations of $f(x)$.*

This theorem precisely shows that when we define a computable function of a computable variable described by a Turing machine, no information will be read off of the

machine given as input, this means that a computable function of a computable real cannot be defined by defining a function that manipulates Turing machines independently of the underlying real numbers that are encoded by those machines.

To each algorithmic description of groups, we can associate an equivalent of the Kreisel-Lacombe-Schoenfield-Ceitin Theorem, which would state that what can be computed from Turing machines that encode the groups could also be computed with descriptions of groups given by oracles. And of course, such theorems should be relativized to different classes of groups. We do not expect that obtaining such theorems will be an easy task, as the Kreisel-Lacombe-Schoenfield-Ceitin Theorem itself is non trivial, and new difficulties seem to arise when trying to establish it for different algorithmic descriptions of groups (see [Rau21b]).

We thus state the following problem:

Problem 19. Establish equivalents of the Kreisel-Lacombe-Schoenfield-Ceitin Theorem for different types of algorithmic descriptions of groups, such as recursive presentations, word problem algorithms, finite presentations together with word problem algorithms, etc. (Or prove that those theorems fail.)

2.3. **Practical implementation.** We stated in the introduction that the purpose of the study of decision problems for groups was to quantify the complexity of different classes of groups. This of course left out an obvious application of the study of decision problems for groups: actual implementation of the solutions to some decision problems.

We want to point out here that describing groups thanks to algorithmic descriptions does not hinder the practical implementation of solutions to decision problems.

A possible critic is that an algorithm which takes as input groups described by descriptions that are stronger than mere finite presentations will not be useful in practice without a way of effectively producing those stronger descriptions, when given only finite presentations. For instance, one might think that a result obtained for groups given by finite presentations together with word problem algorithms will not be useful in practice unless we are set in a class of groups with uniformly solvable word problem, but in this last case the study of groups given by descriptions that consist in finite presentations together with word problem algorithms does not add anything to the study of groups given by finite presentations.

This reproach is related to the central role that finite presentations play in the study of decision problems for groups, but it is not fundamental.

Of course, if some decision problem is implemented at the expressed purpose of launching a blind search for groups with certain properties, using random finite presentations, then surely the implementing of algorithms defined on non-recursively enumerable sets will be a problem. See 2.4 for more about this.

On the other hand, if an algorithm is meant to be used on known groups, to answer questions that might still be open about those groups (as in [NT15], where an algorithm that detects Property (T) is used), then the use of algorithmic descriptions that differ from finite presentations is not a problem.

The step that goes from an abstract isomorphism class of groups to a first effective description of it cannot be deemed "computable" or not. Those description appear through the work of mathematicians confronted to groups that stem from different areas of mathematics. In topology, the description that is most easily obtainable of the fundamental group of some manifold is often a finite presentation. But this is not always the case. The Thompson group F can be seen as a group of piecewise affine homeomorphisms, whose composition can readily be computed, thus giving a word problem algorithm for it, whereas, at first glance, there is no reason to believe that this group is finitely presented. This gives an example of a group for which one obtains much more naturally a solution to the word problem than a finite presentation. Thus there is no reason to believe that finite presentations are the only descriptions that can be used for actual implementation of decision problems. And in many cases, when a finitely presentable group is not yet described by a finite presentation, the most direct solution to finding a finite presentation for it uses the description of a normal form for its elements, which gives a solution to the

word problem in this group. In this case, it is very natural to give to a program that should find properties about this group all the information we have on it.

2.4. **The formalism of** [GMW12]**.** We finally address a remark that comes from [GMW12]. In that article, the authors introduce a notion of "a property recursive modulo the word problem" which differs from the notion of "a property that can be recognized by an algorithm that takes as input both a finite presentation and a word problem algorithm for the group".

Indeed, their definition can be rephrased as follows:

Definition 20 ([GMW12], Definition 2.4)**.** A class of finitely presented groups $\mathcal{C}$ is said to be recursive modulo the word problem if, whenever $\mathcal{D}$ is a set of presentations in which the word problem is uniformly solvable, there exist r.e. sets of groups $\mathcal{X}$ and $\mathcal{Y}$ such that $\mathcal{C} \cap \mathcal{D} \subseteq \mathcal{X}$, $\mathcal{D} \setminus \mathcal{C} \subseteq \mathcal{Y}$, $\mathcal{C} \cap \mathcal{D} \cap \mathcal{Y} = \emptyset$ and $(\mathcal{C} \setminus \mathcal{D}) \cap \mathcal{X} = \emptyset$.

Here, when we say that a set of finitely presented groups $\mathcal{C}$ is r.e., we of course mean that there exists an effective enumeration of finite presentations which contains at least a presentation for each group in $\mathcal{C}$.

A problem of this definition is the fact that the sets $\mathcal{X}$ and $\mathcal{Y}$ are to be r.e.. There exist families of finitely presented groups with uniformly solvable word problem that are not r.e., and even, that are not contained in any r.e. set of finitely presented groups with uniformly solvable word problem. This follows from the following results.

Lemma 21. *Let $\mathcal{C}$ be a set of finitely presented groups in which the word problem is uniformly solvable. If $\mathcal{C}$ is r.e., then there exists a recursive function which is a common asymptotic upper bound to the time complexity for the word problem for groups in $\mathcal{C}$.*

Proof. The argument is a diagonal argument: if G_1, G_2,... is an enumeration of the groups in $\mathcal{C}$, the function f which to n associates the longest time it took, in the first n groups of the enumeration, to determine whether or not a word of length at most n defines the identity, is recursive, and it bounds asymptotically the time complexity of the groups in $\mathcal{C}$.

One could also invoke the fact that the Higman-Neumann-Neumann Theorem which states that any countable group embeds in a two generated group is known to preserve solvability of the word problem, and thus for a class $\mathcal{C}$ of groups as in the statement of the lemma, there must exist a single finitely generated group with solvable word problem in which all the groups of $\mathcal{C}$ embed. The time complexity of the word problem in this group is an asymptotic upper bound for the time complexity of the word problem for any group that belongs to $\mathcal{C}$. □

Using this lemma with the work of Kharlampovich, Myasnikov and Sapir from [KMS17], we obtain the following corollary:

Corollary 22. *The set of finitely presented residually finite groups is not r.e., and it cannot be contained in a r.e. set of finitely presented groups with uniformly solvable word problem.*

Proof. A well known result of McKinsey ([McK43]) shows that the word problem is uniform on all finitely presented residually finite groups. However, in [KMS17], it is shown that for any recursive function f, there exists a finitely presented residually finite group whose word problem time complexity is more than f. The result then follows from Lemma 21. □

Now consider the consequences of this corollary on Definition 20: to prove that a set $\mathcal{C}$ is "recursive modulo the word problem", applying the definition with $\mathcal{D}$ being the set of residually finite groups, one needs to find r.e. sets $\mathcal{X}$ and $\mathcal{Y}$ that, in particular, must cover $\mathcal{D}$. But then the word problem cannot be uniformly solvable in $\mathcal{X} \cup \mathcal{Y}$, and thus one ends up working in sets where the word problem algorithm is not solvable to show that something is recursive modulo the word problem.

One could deal with this problem by imposing in Definition 20 that the set $\mathcal{D}$ be r.e.. We do not recommend such a definition either.

Remember our analogy with computable analysis: the notion of computability that we are interested in is an equivalent of what is known as Markov computability, (or computability "according to the Russian school", see [PDD16]): a computable function defined on computable reals takes as input a Turing machine that defines a computable real and produces a Turing machine that defines the image of this number.

On the other hand, considering that a problem is solvable if and only if it is solvable on every r.e. set of groups (or r.e. set of finite presentations) is a definition similar to that of Banach-Mazur computability: a function is Banach-Mazur computable if and only if it maps every computable sequence of computable reals to a computable sequence of computable reals. This notion was first developed by Banach and Mazur between 1936 and 1939, but their results were not published at that time. The result of their work was written down later in [Maz63], see [Her01] for a more modern point of view.

Banach-Mazur computability was shown to differ from Markov computability ([Her02]), and other notions were preferred to it (precisely because the example of [Her02] of a Banach-Mazur computable function that is not Markov computable can be considered "too" pathological). We likewise prefer Markov computability to Banach-Mazur computability for decision problems for groups.

While we expect most classes of groups in which many decision problems are solvable to be r.e. (all known sets of finitely presented groups on which the isomorphism problem is solvable are r.e.), we should not dismiss "by definition" the possibility of solving interesting problems on non r.e. sets of groups.

We stress the fact that defining a decision problem on a non-r.e. set is a common thing, the question "does there exist an algorithm that, for groups in $\mathcal{C}$, computes ...?" is valid even when the set $\mathcal{C}$ is not r.e.: the simple statement of computable analysis that the function $x \mapsto 3x$ is Markov computable asserts solvability of a problem defined on a non-r.e. set (as was remarked by Turing in 1936, [Tur37]).

3. Recognizable groups

We now make a few remarks about the concept of group recognizability, that respond both our remark on the need of investigating decision problems for marked groups, and the one that concerns the need to investigate decision problems for groups described by stronger descriptions than finite presentations.

A group G is *recognizable* with respect to a certain type of description if there is an algorithm which takes as input a description of the selected type, and decides whether or not it is a description of G. This definition holds whether G is a marked group or an abstract group. We will thus talk about *abstract recognizability* and *marked recognizability.*

We call a group G (abstract or marked) *semi-recognizable* with respect to a type of description when there is an algorithm that stops exactly on the descriptions of this type that define G.

We will see that marked recognizability is not an isomorphism invariant: a group can be recognizable with respect to some marking, while not for others.

Let us first recall that the first result stated in this article, Proposition 6, implies the following:

Proposition 23. *No group, abstract or marked, is recognizable from the r.e. algorithm description.*

The trivial group is the only semi-recognizable group from the r.e. algorithm description (also either as a marked group, or as an abstract group).

The situation is quite different for finitely presented groups and for other strong descriptions. We discuss this now.

3.1. From finite presentations.

The following is a well known result that follows from the use of Tietze transformations. A proof of it already appeared in this paper (Proposition 9).

Proposition 24. *For groups described by finite presentations, every group, marked or abstract, is semi-recognizable. What's more, the method of recognition is uniform on all finite presentations.*

Remark 25. As we are trying to solve the isomorphism problem using descriptions that differ from finite presentations, one of the aspects that appears to be challenging and of foremost importance is to obtain group descriptions and classes of groups for which the result of this proposition holds.

Let us explain why.

In order to solve the isomorphism problem for a certain class $\mathcal{C}$ of groups using finite presentations, it suffices to be able to give a list of presentations with exactly one presentation for each isomorphism class of groups in $\mathcal{C}$ (this was first explained in detail in [Mos73]). This very natural way to solve the isomorphism problem *does not work anymore* when dealing with descriptions of groups from which the groups are not uniformly semi-recognizable. For instance, it is easy to list a sequence of Word Problem Algorithms which contains exactly one copy of a description of every finitely generated abelian group, using the classification theorem for finitely generated abelian groups. This does not allow one to solve the isomorphism problem for abelian groups given by word problem algorithms (since this isomorphism problem is unsolvable, see Subsection 1.3.3). On the contrary, using this classification theorem to list a set of finite presentations which is in bijection with the isomorphism classes of finitely generated abelian groups *does* constitute a solution to the isomorphism problem for abelian groups given by finite presentations (and this, not even taking into consideration whether or not the proof of the theorem is effective). The former case is very unsatisfactory: we cannot ask for a better classification theorem than that of finitely generated abelian groups, and still it does not translate into a solution to the isomorphism problem for abelian groups given by their word problem algorithm.

The solution to the isomorphism problem in a class $\mathcal{C}$ of groups should constitute a step towards obtaining a classification theorem for groups in $\mathcal{C}$, and looking for ways to solve the isomorphism problem in already classified classes of groups is probably not something one would be inclined to spend much time on.

This emphasizes the need for finding descriptions of groups for which semi-recognizability results can be established. In particular, the algorithmic generalizations of finite presentations we discuss in Section 4 would satisfy this property.

We now return to discussing recognizability for finitely presented groups.

Some finitely presented groups admit markings which are recognizable. For instance, notice that given a finite presentation π over n generators, while it is well known that we cannot decide whether π defines a free group, we can easily decide whether it defines a free group of rank n: it suffices to check whether or not π has relations. This shows that free groups are recognizable with respect to any basis. Also, cyclic groups defined by presentations on one generator are recognizable, since any presentation on a single generator a can be simplified to a presentation of the form $\langle a|a^n\rangle$ for some integer n. However, we have the following proposition which shows that this is a rare situation.

Proposition 26. *Consider a finitely presented group G, marked by a generating family S. Add two generators symbols u and v to S, which correspond to the identity in G. Then G is not recognizable from finite presentations with respect to this new marking. And G is not abstractly recognizable.*

Proof. This proof follows from the construction given in [Mil92] which is used to prove the Adian-Rabin theorem. The theorem cannot be applied directly, since "being isomorphic to G" is not always a Markov property, and because the Adian-Rabin theorem is stated only for abstract group properties.

We only use the fact that, in [Mil92], given a finitely presented group H with unsolvable word problem, a family of finite presentations π_w, $w \in H$, is constructed, such that: π_w can be effectively constructed from w, and π_w defines the trivial group if and only if $w = 1$ in H. What's more, π_w is defined on two generators u and v.

It is then easy to see that for a given group G as in the statement of this proposition, the presentation for the free product of G and of the group defined by π_w defines a marking of G with respect to the enlarged generating family if and only if $w = 1$ in H, and that, by Grushko's theorem on the rank of a free product, G and this free product are abstractly isomorphic also if and only if $w = 1$. □

This result could surely be improved, as we conjecture that only the aforementioned groups (free groups, cyclic groups and the trivial group) admit markings which are recognizable from finite presentations. However, to obtain such results, one would have to encode undecidability of some sort with much less space to work with than the one given by two generators fully dedicated to encoding something uncomputable. The easiest open example may be to prove that, given a presentation on two generators a and b, one cannot decide whether or not it is a presentation of $\mathbb{Z}$ on the generating set $\{1; 0\}$.

Problem 27. Characterize the marked groups that are recognizable from the finite presentation description, and fill the "two additional generators" gap left by the proof of the Adian-Rabin Theorem.

There is, however, one case for which the situation is clear, that is for groups with unsolvable word problem:

Proposition 28. *Any marking of a finitely presented group with unsolvable word problem is unrecognizable from the finite presentation description.*

Proof. The proof is a variation on the solution to the word problem for simple groups ([Kuz58]). Consider a generating family S of G, and a finite presentation of G over S. To know whether a word w on S is the identity in G, add w to the relations of G. Then $w = 1$ in G if and only if G is isomorphic, as a marked group, to the group defined by this new presentation. □

Note that this proposition holds more generally for any group description for which "adding a relation" is a recursive operation. This remark allows us to comment on the link between recognizability as a marked group, and as an abstract group. Indeed, we ask:

Problem 29. Is there a natural example of a group G, recognizable as an abstract group, while some of its markings are not recognizable? (in some class $\mathcal{C}$ of groups, with respect to some type of description.)

This question is very much related to a question of Kharlampovich and Sapir (in [KS95]): can a variety of groups have solvable isomorphism problem (for groups given by presentations), while still containing a group with unsolvable word problem? In a class of groups where the isomorphism problem is solvable, any group is abstractly recognizable, thus by Proposition 28, a positive answer to that question would give a positive answer to Problem 29. It was also remarked in [KS95] that for a Hopfian group G with unsolvable word problem, the marked unrecognizability of Proposition 28 translates into unrecognizability as an abstract group.

3.2. **From finite presentations together with word problem algorithms.** The next proposition gives further link between solvability of the word problem and marked recognizability.

Proposition 30. *Finitely presented groups with solvable word problem are recognizable, as marked groups, and with respect to any marking, when described by finite presentations together with word problem algorithms.*

Proof. The proof is straightforward, it suffices to check whether each group satisfies the relations of the other one using the word problem algorithm. □

The marked groups that are recognizable from the description "finite presentation-word problem algorithm" are thus easily completely classified. On the contrary, which groups are abstractly recognizable from this description is an interesting open problem.

Indeed, we have already mentioned in the introduction the results from [GW09] and [GMW12], which provide examples of recognizable groups, including free groups and more generally limit groups.

On the other hand, Miller, in [Mil72, Theorem 26, Chapter IV], has built a group that is unrecognizable in the class of finitely presented residually finite groups. As, by [McK43], finitely presented residually finite groups have uniformly solvable word problem, this group cannot, in the class of all finitely presented groups with solvable word problem, be recognizable from the description constituted of a finite presentation together with a word problem algorithm.

This shows that the problem of abstract group recognition from finite presentations together with word problem algorithms has a non-trivial answer. Miller's group has unsolvable conjugacy problem, and thus it bears some undecidability itself. We are not able to tell whether we should expect many groups to be recognizable from this description, or on the contrary that Miller's example be the norm, and recognizable groups be the exceptions.

Note that there are several issues one will encounter when trying to extend the methods of [GW09]. For instance, if one were to attempt to extend the results of this article to hyperbolic groups, one will be faced with the problem that it is not known whether the universal theory of every hyperbolic group is decidable, whereas the results from [GW09] relied on the fact that the universal theory of free groups is decidable.

We sum this up in the following problem, whose investigation we expect to be fruitful.

Problem 31. Describe the groups that are abstractly recognizable from the description that consists of a finite presentation together with a word problem algorithm.

4. Relative marked quotient algorithms

We've seen in Section 1 that giving a finite presentation of a marked group G is equivalent to giving a pair of algorithms, the r.e. algorithm for G, and a marked quotient algorithm that recognizes r.e. algorithms for marked quotients of G. But as algorithmic problems for groups are often set in restricted classes of groups, one in fact seldom needs the full strength of the marked quotient algorithm of a group. Because of this, we here introduce *relative marked quotient algorithms,* and we will see that, for some classes of groups, the relative marked quotient algorithms are not associated to finite presentations.

4.1. **Definitions and Basic properties.**

4.1.1. *Definitions.* We've seen already, in the first section of this paper, that there are two natural marked quotient algorithms that finitely presented groups admit, the first one characterizes them amongst r.p. groups.

(1) The *marked quotient algorithm* A_Q^G of a marked group G, which takes as input a r.e. algorithm for a group, and stops if and only if the input is a marked quotient of G.
(2) The *WPI marked quotient algorithm* (for Word Problem as Input), which we note $A_{Q_{WP}}^G$, which takes as input a word problem algorithm for a group, and decides whether or not this group is a marked quotient of the starting group.

Definition 32. Let $\mathcal{C}$ be a class of groups. We say that an algorithm $A_{Q,\mathcal{C}}^G$ is *a $\mathcal{C}$-marked quotient algorithm* for a marked group G if it takes as input r.e. algorithms for groups in $\mathcal{C}$ and stops exactly on marked quotients of G.

We say that an algorithm $A_{Q_{WP},\mathcal{C}}^G$ is *a WPI $\mathcal{C}$-marked quotient algorithm* if it takes as input word problem algorithms for groups in $\mathcal{C}$, and decides whether or not they define marked quotients of $\mathcal{C}$. If G admits such an algorithm, we say that G *has computable quotients in* $\mathcal{C}$ (as in [Rau21a]).

Note that in [Rau21a], we gave an example of a residually finite group with computable finite quotients, (as defined above, with $\mathcal{C}$ being the set of finite groups), which is not recursively presented. This shows that the WPI marked quotient algorithm does not necessarily come from a marked quotient algorithm together with a recursive presentation.

The proof of Lemma 10, which asserted that having a marked quotient algorithm is independent of a given generating family, can easily be extended to relative quotient algorithms.

Lemma 33. *If a group G admits a $\mathcal{C}$-marked quotient algorithm (or a WPI $\mathcal{C}$-marked quotient algorithm) with respect to some marking, then it admits $\mathcal{C}$-marked quotient algorithms (respectively, WPI $\mathcal{C}$-marked quotient algorithms) with respect to any marking.*

Note that in our definition, we supposed that the class $\mathcal{C}$ is a class of groups, i.e. a class of marked groups which is closed under abstract isomorphism. We could also consider marked quotient algorithms relative to sets of marked groups. In this case, the previous lemma may fail. We do not investigate marked quotient algorithms relative to sets of marked groups here.

Remark 34. The terminology used in [Rau21a] to investigate finite quotient algorithms is different from the one we are using here. Indeed, in [Rau21a], a marked group G is said to have "recursively enumerable finite quotients" if it is possible to recognize a marked finite quotient of G, *when this quotient is described by a finite presentation.* It is easy to see that this is equivalent to having a finite quotient algorithm, in the sense of Definition 32, because all finite groups are finitely presented, and because the set of finite presentations for finite groups is recursively enumerable.

Furthermore, we also noted in [Rau21a] that having "recursively enumerable finite quotients" was equivalent to being able to list all finite presentations for marked finite quotients of G. This equivalence also relies on the fact that the set of finite presentations of finite groups is recursively enumerable.

Because we want to be able to investigate marked quotient algorithms for sets of groups which are not necessarily all finitely presented, and which may not be recursively enumerable (whether by finite or recursive presentations), the definitions we propose here are to be preferred.

However, note that in other classes of groups, the marked quotient algorithm could be equivalently defined as an algorithm that enumerates a set of finite presentations, for instance in the class of hyperbolic groups.

4.1.2. *Basic Properties.* In what follows, $\mathcal{C}$ is a class of groups, and $\mathcal{QC}$ denotes either the set of groups which admit a $\mathcal{C}$-marked quotient algorithm, or the set of groups which admit a WPI $\mathcal{C}$-marked quotient algorithm.

The following propositions are straightforward.

Proposition 35. *$\mathcal{QC}$ contains all finitely presented groups. More generally, it is stable by quotients by subgroups that are finitely generated as normal subgroups (i.e. adding finitely many relations to a group).*

Proposition 36. *$\mathcal{QC}$ is stable by free and direct products.*

Corollary 37. *$\mathcal{QC}$ is stable by HNN-extensions or amalgamated products over finitely generated groups. More generally the fundamental group of a graph of groups with vertex groups in $\mathcal{QC}$ and finitely generated edge groups is again in $\mathcal{QC}$.*

For a class $\mathcal{C}$ of groups, call a group G *residually* $\mathcal{C}$ if every non-trivial element of G has a non-trivial image in a group of $\mathcal{C}$. Note $\mathcal{RC}$ the class of residually $\mathcal{C}$ groups. Any group G has a greatest quotient in $\mathcal{RC}$, namely the quotient of G by the intersection of all normal subgroups N of G for which G/N is in $\mathcal{C}$. This group is called the *residually $\mathcal{C}$ image* of G, and is noted $G_{\mathcal{C}}$ (as in [Dys74]).

A residually $\mathcal{C}$ group is called *finitely presented as a residually $\mathcal{C}$ group* if it is the residually $\mathcal{C}$ image of a finitely presented group.

Proposition 38. *A group G admits a $\mathcal{C}$-marked quotient algorithm if and only if the group $G_{\mathcal{C}}$ admits such an algorithm. Thus a group G which is finitely presented as a residually $\mathcal{C}$ group admits a $\mathcal{C}$-marked quotient algorithm.*

Proof. This follows from the universal property of $G_{\mathcal{C}}$: any morphism from G to a group in $\mathcal{C}$ factors through $G_{\mathcal{C}}$. □

This proposition gives rise to many groups which have computable quotients in $\mathcal{C}$, sometimes for the sole reason that they do not have any quotients in $\mathcal{C}$. For instance, a simple group admits marked quotient algorithms in every class of groups to which it does not belong.

Note how this relate to the notion of *equationally noetherian groups* (introduced in [BMR99], we quote the definition of [GW18]): a group G is called *equationally noetherian* if for every finitely generated group H, there exists a finitely presented group $\hat{H}$ and an epimorphism $\rho : \hat{H} \to H$ such that every morphism from $\hat{H}$ to G factors through ρ. It follows from that definition that for G an equationally noetherian group, every finitely generated group admits a $S(G)$-marked quotient algorithm, where $S(G)$ designates the set of subgroups of G.

4.2. **Extending classical uses of marked quotient algorithms to relative algorithms.** We quote three possible uses of marked quotient algorithms, by recalling instances of already existing proofs where finite presentations were used mostly for their marked quotient algorithm component, and discuss how to extend them to marked quotient algorithms relative to some classes of groups.

(1) The Tietze transformations algorithm ([Tie08], see [LS77]);
(2) McKinsey's algorithm ([McK43]);
(3) Pickel's method (named in [KS95], from [Pic71]).

4.2.1. *Tietze's Transformations.* This first point was already discussed in Sections 1 and 3. We noted there that finitely presented groups are uniformly semi-recognizable, and that this property is important because it allows us to avoid the embarrassing situation of having a completely understood and classified set of groups with unsolvable isomorphism problem. This result, usually explained by Tietze transformations, is immediately explained in terms of marked quotient algorithms: if two marked groups are each a quotient of the other, they are isomorphic.

To be able to use relative marked quotient algorithms to obtain semi-recognizability results, one needs to be able to find some groups in a class $\mathcal{C}$ that admit a $\mathcal{C}$-marked quotient algorithm. This is summed up in the following straightforward proposition:

Proposition 39. *Groups in a class $\mathcal{C}$ are uniformly semi-recognizable, either as marked groups, or abstractly, from the description that consists in a recursive presentation together with a $\mathcal{C}$-marked quotient algorithm.*

While, from the three results mentioned above, this proposition might not be the one that yields the most impressive results, it is in fact the most important one. We later discuss how it can apply to groups finitely presented inside varieties.

4.2.2. *McKinsey's Algorithm.* The second point is McKinsey's algorithm. The statement of McKinsey's theorem is the following (we only state it for groups, it was originally stated for *finitely reducible algebras*):

Theorem 40. *(McKinsey,* [McK43]*) A finitely presented residually finite group has solvable word problem.*

Recall that the proof of this theorem is as follows. All that we need to prove is that a finitely presented residually finite group is co-r.p.. Let G be a finitely presented residually finite group. Let w be a word on the generators of G. Enumerate all finite groups by listing all possible Cayley tables. For each obtained finite group, decide whether it is a quotient of G by checking whether the finitely many relations of G hold in it. Then, if it is indeed a quotient of G, and if the image of w in this quotient is a non-identity element, then answer that $w \neq 1$ in G. If w is indeed a non-identity element of G, by definition of "being residually finite", this process terminates.

It is then easy to see that the most general formulation of this argument is the following:

Theorem 41. *Suppose that there exists a recursive enumeration by word problem algorithms of a class $\mathcal{C}$ of groups (and thus that $\mathcal{C}$ consists only of groups with solvable word*

problems). Then any residually $\mathcal{C}$ group with a $\mathcal{C}$-marked quotient algorithm is co-r.e., and thus any r.e. residually $\mathcal{C}$ group with a $\mathcal{C}$-marked quotient algorithm has solvable word problem.

Furthermore, there exist uniform ways of producing the said co-r.e. and word problem algorithms, from, respectively, a $\mathcal{C}$-marked quotient algorithm or a $\mathcal{C}$-marked quotient algorithm with a recursive presentation.

Several versions of this theorem have already been used to prove that some classes of groups have solvable word problem. For instance, Sela, in [Sel09], used the fact that a finitely presented residually H group, for H some hyperbolic group, must have solvable word problem.

Note also that one can replace residual finiteness in the statement of this theorem by conjugacy separability or subgroup separability *relative to a class* $\mathcal{C}$, and obtain theorems about the conjugacy and membership problems. See [Rau21a] for the usual definitions, which correspond to conjugacy (resp. subgroup) separability relative to the class of finite groups. For instance, subgroup separability with respect to amenable groups was introduced in [GK14] (but the class of amenable groups cannot be used to obtain a McKinsey algorithm). One should replace accordingly the enumeration of $\mathcal{C}$ by word problem algorithms by an enumeration by conjugacy problem algorithms, or by membership problem algorithms.

4.2.3. *Pickel's Method.* The last point corresponds to what is known as Pickel's method. In [Pic71], Pickel showed that only finitely many nilpotent groups can have the same set of finite quotients. From this (and the fact that nilpotent groups are finitely presented), it is easy to deduce that every finitely generated nilpotent group is abstractly recognizable from other nilpotent groups. Indeed, consider a nilpotent group G, and the list G_1, ..., G_l of nilpotent groups with the same set of finite quotients as G. Given a finite presentation of a group H, use the semi-recognition property (Tietze transformations) to try and prove that H appears in the list G, G_1, ..., G_l; all the while, list all finite quotients of H and try to prove that some finite group is a quotient of G and not of H, or of H and not of G. This process always terminates and allows one to decide whether or not a given presentation defines G.

It seems that Pickel's method can be used only with the marked *finite* quotient algorithm. Indeed, not only does it rely on the solution to the isomorphism problem for finite groups, but also on the fact that from the WPI finite marked quotient algorithm of a group G, which tells us whether a given finite *marked group* is a quotient of G, one can obtain an algorithm which decides whether an *abstract* finite group is a quotient of G. This second fact uses both the isomorphism problem for finite groups and the fact that any abstract finite group admits finitely many markings of any given arity. The best directly available generalization of Pickel's method is thus:

Proposition 42. *There is an effective procedure which, given two WPI finite marked quotient algorithms for groups G and H, terminates if and only if their set of finite quotients differ.*

Note that, since Pickel's article, many results of the same kind were obtained for various classes of groups. A class $\mathcal{C}$ of residually finite groups is said to be *profinitely rigid* if any two different groups of $\mathcal{C}$ have different sets of finite quotients. Note that a marked residually finite group is always uniquely defined by its marked finite quotients, and thus the notion of profinite rigidity naturally appears as one moves from the category of marked groups to that of abstract groups. See [Rei19] for a survey on profinite rigidity.

To be able to use notions of rigidity with classes of quotients that are possibly infinite, in order to obtain results akin to the one given above for nilpotent groups, one would have to define "abstract quotient algorithms". For instance, one could consider an algorithm that takes as input a finite presentation and decides whether or not it defines an abstract quotient of a given group G. However, such algorithms will never exist when set in unrestricted classes of groups. For instance, being able to decide whether or not a finite presentation defines an abstract quotient of a rank two free group is equivalent to deciding

whether the rank of this group is less or equal to two, which is impossible by the Adian-Rabin construction (see the proof of Proposition 26). But this does not mean that such algorithms cannot exist in restricted classes of infinite groups, the study of such algorithms could be interesting for residually free groups, residually hyperbolic groups, etc.

4.3. **Group varieties.** There in fact already exists a wide range of classes of groups that satisfy the generalized Tietze transformations criterion: groups that are finitely presented *inside* group varieties. A group variety is a class of groups defined by a set of *laws*, and a law is a universal sentence of the form:

$$\forall x_1, ..., x_n,\ W = 1$$

for W a word on the letters $\{x_1, ..., x_n\} \cup \{x_1^{-1}, ..., x_n^{-1}\}$. A group is said *finitely presented inside a variety* $\mathcal{V}$ if it admits a presentation with finitely many relations apart from the infinitely many relations that constitute the laws of the variety $\mathcal{V}$. This notion in fact corresponds to being finitely presented as a residually $\mathcal{V}$ group, because group varieties are stable under taking subgroups and forming unrestricted direct products, and thus the class of residually $\mathcal{V}$ groups corresponds to $\mathcal{V}$. It follows that a group which admits a finite presentation inside a variety admits a marked quotient algorithm inside this variety. For a variety defined by a recursively enumerable set of laws, this is in fact an equivalence.

Theorem 43. *Let $\mathcal{V}$ be a group variety defined by a r.e. set of laws. A recursively presented group H of $\mathcal{V}$ admits a $\mathcal{V}$-marked quotient algorithm if and only if is finitely presented as a group in the variety $\mathcal{V}$.*

What's more, from a pair $(A_{re}^H, A_{Q,\mathcal{V}}^H)$, a finite presentation in the variety $\mathcal{V}$ of H can be effectively obtained.

Proof. The proof is identical to that of Theorem 3, except that one needs to add the laws that defines $\mathcal{V}$ when building the algorithm $\mathcal{A}_{re}^n$. □

The residually $\mathcal{V}$ image $G_{\mathcal{V}}$ of a group G is the group obtained by imposing on G the laws of $\mathcal{V}$. The previous theorem thus characterizes groups with a $\mathcal{V}$ marked quotient algorithm amongst r.p. groups.

Corollary 44. *A r.p. group G admits a $\mathcal{V}$-marked quotient algorithm if and only if $G_{\mathcal{V}}$ is finitely presented as a group in $\mathcal{V}$.*

This shows that the notion of having a $\mathcal{V}$ marked quotient algorithm is already well known for group varieties.

4.4. **Elementary marked quotient algorithms.** The interest of relative quotient algorithms lies mostly in situations where the marked quotient algorithms are *not* given by finite presentations, and the case of an algorithm that, to decide whether a group H given by an algorithm A_{re}^H is a quotient of a group G, checks whether H satisfies a finite set of relations *which is independent of H* is considered to be the trivial case.

Definition 45. A marked quotient algorithm $\mathcal{A}_Q$ for a marked group H is called *elementary* if there exists a finite set $\mathcal{R}$ of relations, such that $\mathcal{A}_Q$ recognizes the set of marked groups that satisfy the relations in $\mathcal{R}$.

The previous theorem about group varieties -as well as its particular instance in the variety of all groups- thus proves that only elementary marked quotient algorithms exist, relative to group varieties.

A first example of a marked quotient algorithm that is not elementary is the finite quotient algorithm (in fact, the torsion quotient algorithm) of the Lamplighter group.

Proposition 46. *The lamplighter group has a non-elementary finite marked quotient algorithm.*

Proof. The lamplighter group L has a finite marked quotient algorithm. Indeed, it admits the following presentation:

$$\langle a, \varepsilon \,|\, \varepsilon^2,\ \left[\varepsilon, a^{-n}\varepsilon a^n\right], n \in \mathbb{Z}\rangle$$

To see whether a finite group F generated by two elements a_1 and ε_1 is a quotient of it, find a multiple N of the order of a_1 using the r.e. algorithm of F. Then, notice that F is a quotient of L if and only if it is a quotient of the group obtained from L by adding the relation a^N. But the quotient $L/\langle\langle a^N\rangle\rangle$ is in fact the finite wreath product $\mathbb{Z}/N\mathbb{Z}\wr\mathbb{Z}/2\mathbb{Z}$, which admits the finite presentation:

$$\langle a, \varepsilon \,|\, \varepsilon^2,\, a^N,\, \left[\varepsilon, a^{-n}\varepsilon a^n\right], 0 \leq n \leq N\rangle$$

It can be effectively checked whether F is a quotient of this finite group, and thus of L.

Because the lamplighter group is residually finite, to prove that its finite quotient algorithm is not elementary, it suffices to prove that it is not finitely presented as a residually finite group. Indeed, it is easy to check that if a finite number of relations r_1, ..., r_p characterized the finite quotients of L that are generated by two elements a_1 and ε_1, the group defined by the presentation $\langle a_1, \varepsilon_1 | r_1, ..., r_p\rangle$ would have L as its residually finite image.

We thus now prove that L is not finitely presented as a residually finite group.

Suppose that we have a group G, given by a presentation $\langle a_1, \varepsilon_1 | r_1, ..., r_p\rangle$, and a morphism $\phi : G \to L$, defined by $\phi(a_1) = a$ and $\phi(\varepsilon_1) = \varepsilon$, which satisfies that any morphism h from G to a finite group F factors through ϕ. The corresponding diagram is as follows:

$$\begin{tikzcd} G \arrow[r, "\phi"] \arrow[dr, "h_0"'] & L \arrow[d, dashed, "h_1"] \\ & F \end{tikzcd}$$

Since L is a quotient of G, L must satisfy the relations of G, those relations must thus be consequences of a finite number of the relations of L. In particular, there must be a natural number N such that the first N relations of L imply those of G. Consider the group H given by the presentation

$$\langle a, \varepsilon \,|\, \varepsilon^2,\, \left[\varepsilon, a^{-n}\varepsilon a^n\right], n \leq N\rangle.$$

The property of G, that all its finite quotients come from quotients of L, must be shared by H: any morphism h_0 from H to a finite group F defines a morphism h_1 from G to F, which by the property of G, factors through ϕ.

$$\begin{tikzcd} G \arrow[r, "\phi_1"] \arrow[drr, "h_1"'] & H \arrow[r, "\phi_2"] \arrow[dr, "h_0"] & L \arrow[d, dashed, "h_2"] \\ & & F \end{tikzcd}$$

To end the proof, we find a finite group which satisfies the relations of H, but not that of L. We define a subgroup of the group $\mathfrak{S}_{5N}$ of permutations on $\{1, ..., 5N\}$. Consider the element σ_0 of $\mathfrak{S}_{5N}$, defined by the following formula:

$$\sigma_0(i) = \begin{cases} i+2 & i \leq 5N-2 \\ i+2-5N & i \geq 5N-1 \end{cases}.$$

Let σ_1 be the product of the transpositions $(1, 2)$ and $(2N+4, 2N+5)$. It is then easy to see that the following relations hold between σ_0 and σ_1:

$$\sigma_1^2 = id,$$
$$\left[\sigma_1, \sigma_0^{-n}\sigma_1\sigma_0^n\right] = id,\ 1 \leq n \leq N,$$
$$\left[\sigma_1, \sigma_0^{-N-1}\sigma_1\sigma_0^{N+1}\right] \neq id.$$

The subgroup of $\mathfrak{S}_{5N}$ generated by σ_0 and σ_1 is thus a finite quotient of the group H, but not of L. This contradicts the supposition that L be finitely presented as a residually finite group. □

The fact that the lamplighter group admits a finite quotient algorithm also follows from a more general result, due to Mostowski ([Mos66]):

Proposition 47. *A group that is finitely presented in a variety $\mathcal{V}$ defined by finitely many laws has a finite marked quotient algorithm (and a WPI finite marked quotient algorithm).*

Proof. From the r.e. algorithm of a finite group, arbitrarily good upper bounds on its cardinality can be found. Thus to decide whether a finite group satisfies a law, only finitely many relations need to be checked. □

We have already seen that any group which satisfies the hypotheses of this proposition admits an elementary $\mathcal{V}$-marked quotient algorithm. However, the marked quotient algorithm given by Mostowski's proposition is not elementary: depending on the size of the input group, the number of relations that this algorithms tests may vary, and the set of relations to test is not fixed *a priori*.

These examples of non-elementary finite quotient algorithms rely on the fact that the considered groups do not belong to the class of finite groups. This thus leaves the following important problem unanswered:

Problem 48. Find a class $\mathcal{C}$ of r.p. groups, and r.p. groups in $\mathcal{C}$, that admit non-elementary $\mathcal{C}$-marked quotient algorithms.

This would constitute a genuine algorithmic generalization of the notion of finite presentation.

References

[Bar03] Laurent Bartholdi. Endomorphic presentations of branch groups. *Journal of Algebra*, 268(2):419–443, oct 2003. 16

[Bau93] Gilbert Baumslag. *Topics in Combinatorial Group Theory*. Birkhäuser Basel, 1993. 9

[BCR19] Iva Bilanovic, Jennifer Chubb, and Sam Roven. Detecting properties from descriptions of groups. *Archive for Mathematical Logic*, 59(3-4):293–312, aug 2019. 16

[BELS19] Martin R. Bridson, David M. Evans, Martin W. Liebeck, and Dan Segal. Algorithms determining finite simple images of finitely presented groups. *Inventiones mathematicae*, 218(2):623–648, may 2019. 5

[BK89] Leonid A. Bokut and Georgii P. Kukin. Unsolvable algorithmic problems for semigroups, groups, and rings. *Journal of Soviet Mathematics*, 45(1):871–911, apr 1989. 9

[BMR99] Gilbert Baumslag, Alexei Myasnikov, and Vladimir Remeslennikov. Algebraic geometry over groups I. Algebraic sets and ideal theory. *Journal of Algebra*, 219(1):16–79, sep 1999. 26

[Bri02] Martin R. Bridson. The geometry of the word problem. In *Invitations to geometry and topology*, pages 29–91. Oxford: Oxford University Press, 2002. 2

[BW15] Martin R. Bridson and Henry Wilton. The triviality problem for profinite completions. *Inventiones mathematicae*, 202(2):839–874, feb 2015. 5

[Cei67] Gregory S. Ceitin. Algorithmic operators in constructive metric spaces. pages 1–80, 1967. 18

[CG05] Christophe Champetier and Vincent Guirardel. Limit groups as limits of free groups. *Israel Journal of Mathematics*, 146(1):1–75, dec 2005. 8

[Deh11] M. Dehn. Uber unendliche diskontinuierliche gruppen. *Mathematische Annalen*, 71(1):116–144, mar 1911. 2

[Deh87] Max Dehn. On infinite discontinuous groups. In *Papers on Group Theory and Topology*, pages 133–178. Springer New York, 1987. 2, 8

[DF19] A.S. Detinko and D.L. Flannery. Linear groups and computation. *Expositiones Mathematicae*, 37(4):454–484, dec 2019. 16

[Dyc82] Walther Dyck. Gruppentheoretische studien. *Mathematische Annalen*, 20(1):1–44, mar 1882. 2

[Dyc83] Walther Dyck. Gruppentheoretische studien. II. Ueber die zusammensetzung einer gruppe discreter operationen, uber ihre primitivitat und transitivitat. *Mathematische Annalen*, 22(1):70–108, mar 1883. 2

[Dys74] Verena Huber Dyson. A family of groups with nice word problems. *J. Aust. Math. Soc*, 17(4):414–425, jun 1974. 25

[EPC+92] David B. A. Epstein, M. S. Paterson, J. W. Cannon, D. F. Holt, S. V. Levy, and W. P. Thurston. *Word Processing in Groups*. A. K. Peters, Ltd., 1992. 16

[GI09] Rostislav I. Grigorchuk and Sergei V. Ivanov. On Dehn functions of infinite presentations of groups. *Geometric and Functional Analysis*, 18(6):1841–1874, mar 2009. 17

[GK57] Joseph R. Shoenfield Georg Kreisel, Daniel Lacombe. *Constructivity in mathematics: Proceedings of the Colloquium held at Amsterdam 1957*. North-Holland Publishing Company, 1957. 18

[GK14] Yair Glasner and Daniel Kitroser. A note on LERF groups and generic group actions. 2014. 27

[GMW12] Daniel Groves, Jason Fox Manning, and Henry Wilton. Recognizing geometric 3-manifold groups using the word problem. 2012. 1, 7, 10, 20, 24

[GW09] Daniel Groves and Henry Wilton. Enumerating limit groups. *Groups, Geometry, and Dynamics*, pages 389–399, 2009. 7, 24

[GW18] Daniel Groves and Henry Wilton. The structure of limit groups over hyperbolic groups. *Israel Journal of Mathematics*, 226(1):119–176, may 2018. 26

[Her01] Peter Hertling. Banach-mazur computable functions on metric spaces. In *Computability and Complexity in Analysis*, pages 69–81. Springer Berlin Heidelberg, 2001. 21

[Her02] Peter Hertling. A Banach-Mazur computable but not Markov computable function on the computable real numbers. In *Automata, Languages and Programming*, pages 962–972. Springer Berlin Heidelberg, 2002. 21

[KMS17] Olga G. Kharlampovich, Alexei Myasnikov, and Mark V. Sapir. Algorithmically complex residually finite groups. *Bulletin of Mathematical Sciences*, 7(2):309–352, mar 2017. 20

[KS95] Olga G. Kharlampovich and Mark V. Sapir. Algorithmic problems in varieties. *International Journal of Algebra and Computation*, 05(04n05):379–602, aug 1995. 16, 23, 26

[Kuz58] Alexander V. Kuznetsov. Algorithms as operations in algebraic systems. *Izv. Akad. Nauk SSSR Ser. Mat.*, 1958. 11, 23

[Loc81] Jody Lockhart. Decision problems in classes of group presentations with uniformly solvable word problem. *Archiv der Mathematik*, 37(1):1–6, dec 1981. 5, 10, 16, 18

[LS77] Roger C. Lyndon and Paul E. Schupp. *Combinatorial group theory*, volume 89. Springer-Verlag, Berlin, 1977. 26

[Mal61] Anatolii I. Maltsev. Constructive algebras I. *Uspehi Mat. Nauk*, 1961. 2

[Mal71] Anatolii I. Maltsev. *The metamathematics of algebraic systems, collected papers: 1936-1967*. North-Holland Pub. Co, Amsterdam, 1971. 2

[Man82] Avinoam Mann. A note on recursively presented and co-recursively presented groups. *Bulletin of the London Mathematical Society*, 14(2):112–118, mar 1982. 14

[Man02] Jason Fox Manning. Algorithmic detection and description of hyperbolic structures on closed 3–manifolds with solvable word problem. *Geometry & Topology*, 6(1):1–26, jan 2002. 7

[Mar54] A. A. Markov. On the continuity of constructive functions (Russian). *Uspekhi Mat. Nauk (N.S.)*, 9:226–230, 1954. 16

[Maz63] S Mazur. Computable analysis, as rozprawy matematyczne impan (dissertationes mathematicae) vol. 33. *Panstwowe Wydawnictwo Naukowy, Warsaw*, 1963. 21

[McK43] J. C. C. McKinsey. The decision problem for some classes of sentences without quantifiers. *Journal of Symbolic Logic*, 8(3):61–76, sep 1943. 9, 20, 24, 26

[Mil72] Charles F. Miller. *On Group-Theoretic Decision Problems and Their Classification. (AM-68)*. Princeton University Press, 1972. 7, 9, 24

[Mil92] Charles F. Miller. Decision problems for groups — survey and reflections. In *Mathematical Sciences Research Institute Publications*, pages 1–59. Springer New York, 1992. 3, 9, 14, 22

[Mos66] A. Mostowski. On the decidability of some problems in special classes of groups. *Fundamenta Mathematicae*, 59(2):123–135, 1966. 29

[Mos73] A. Wlodzimierz Mostowski. Uniform algorithms for deciding group-theoretic problems. In *Word Problems - Decision Problems and the Burnside Problem in Group Theory*, pages 525–551. Elsevier, 1973. 22

[Nie07] André Nies. Describing groups. *Bulletin of Symbolic Logic*, 13(3):305–339, sep 2007. 16

[NT15] Tim Netzer and Andreas Thom. Kazhdan's property (T) via semidefinite optimization. *Experimental Mathematics*, 24(3):371–374, jun 2015. 19

[Oza14] Narutaka Ozawa. Noncommutative real algebraic geometry of Kazhdan's property (T). *Journal of the Institute of Mathematics of Jussieu*, 15(1):85–90, jul 2014. 12

[PDD16] Michelle Porter, Adam Day, and Rodney Downey. Notes on computable analysis. *Theory of Computing Systems*, 60(1):53–111, dec 2016. 18, 21

[Pic71] P. F. Pickel. Finitely generated nilpotent groups with isomorphic finite quotients. *Transactions of the American Mathematical Society*, 160:327–327, 1971. 26, 27

[Rab58] Michael O. Rabin. Recursive unsolvability of group theoretic problems. *The Annals of Mathematics*, 67(1):172, jan 1958. 9

[Rab60] Michael O. Rabin. Computable algebra, general theory and theory of computable fields. *Transactions of the American Mathematical Society*, 95(2):341, may 1960. 2

[Rau21a] Emmanuel Rauzy. Computability of finite quotients of finitely generated groups. *Journal of Group Theory*, 0(0), oct 2021. 24, 25, 27

[Rau21b] Emmanuel Rauzy. Computable analysis on the space of marked groups. 2021. 16, 19

[Rei19] Alan W. Reid. Profinite rigidity. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*. World Scientific, may 2019. 27

[Rot99] Joseph J. Rotman. *An Introduction to the Theory of Groups*. Springer New York, 1999. 9

[Sel09] Zlil Sela. Diophantine geometry over groups VII: The elementary theory of a hyperbolic group. *Proceedings of the London Mathematical Society*, 99(1):217–273, feb 2009. 27

[Spe49] Ernst Specker. Nicht konstruktiv beweisbare satze der analysis. *Journal of Symbolic Logic*, 14(3):145–158, sep 1949. 3

[Tie08] Heinrich Tietze. Über die topologischen invarianten mehrdimensionaler mannigfaltigkeiten. *Monatshefte für Mathematik und Physik*, 19(1):1–118, dec 1908. 13, 26

[Tou18] Nicholas W. M. Touikan. Detecting geometric splittings in finitely presented groups. *Transactions of the American Mathematical Society*, 370(8):5635–5704, mar 2018. 7

[Tur37] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937. 21