Bull. Math. Sci. (2017) 7:309–352
DOI 10.1007/s13373-017-0103-z

Algorithmically complex residually finite groups

Olga Kharlampovich[1] · **Alexei Myasnikov**[2] · **Mark Sapir**[3]

Received: 25 October 2016 / Revised: 12 January 2017 / Accepted: 9 March 2017 /
Published online: 20 March 2017

Abstract We construct the first examples of algorithmically complex finitely presented residually finite groups and the first examples of finitely presented residually finite groups with arbitrarily large (recursive) Dehn functions, and arbitrarily large depth functions. The groups are solvable of class 3.

Keywords Word problem · Depth function · Dehn function · Minsky machine

Contents

Communicated by Efim Zelmanov.

O. Kharlampovich: Partially supported by NSF Grant DMS-0700811, A. Myasnikov: Partially supported by NSF Grants DMS-0700811 and DMS-0914773. M. Sapir: Partially supported by NSF Grant DMS-1500180 and by BSF (USA-Israel) Grant 2010295.

✉ Mark Sapir
m.sapir@vanderbilt.edu

Olga Kharlampovich
okharlampovich@gmail.com

Alexei Myasnikov
amiasnik@stevens.edu

1 Department of Mathematics and Statistics, Hunter College, City University of New York, New York, NY 10065, USA

2 Stevens Institute of Technology, Hoboken, NJ 07030, USA

3 Department of Mathematics, Vanderbilt University, Nashville, TN 37240, USA

1 Introduction

1.1 The problem and previous approaches for a solution

It is well known that finitely presented residually finite algebraic systems of finite signature (say, semigroups or groups) are much simpler algorithmically than arbitrary finitely presented algebraic systems. For example, the word problem in every such algebraic system is decidable. In this paper we discuss the question "how complicated finitely presented residually finite algebraic systems can be in the cases of groups and semigroups?".

The most "common" residually finite groups and semigroups, are linear (say, over a field). These groups and semigroups are algorithmically "tame": the word problem there is decidable in polynomial time and even log-space [34]. The Dehn function is a well-known indicator of complexity of the word problems: the smaller the Dehn function the easier the word problem. The converse implication does not hold, however. One can construct groups with decidable word problem and very large Dehn functions [39,59]. However, no examples of finitely presented residually finite groups or even semigroups with superexponential Dehn function are known. Thus one of the main open problems in this area is how large could the Dehn function of a residually finite finitely presented group or semigroup be. The question for groups was known since early 90s. It was open for so long because all known methods to construct algorithmically hard groups produced either non-residually finite groups or groups where the question about their residual finiteness is very difficult. Not much is known even for linear groups (note that Gersten asked [19,20] if there exists a uniform upper bound for Dehn functions of linear groups). Let us briefly discuss the previous attempts to solve the problem and the reasons why these methods did not work.

1.1.1 Method 1. Known groups with large Dehn functions

One could hope that some of the known finitely presented groups with very large Dehn function may turn out to be residually finite, which would shed some light on how to produce residually finite groups with even larger Dehn functions. Unfortunately, this is not the case, all these groups are non-residually finite. For example, the Dehn function of the one relator group $G_{(1,2)} = \langle a, b \mid b^{-1}a^{-1}bab^{-1}ab = a^2\rangle$ introduced by Baumslag in [2] is bigger than any iterated exponent (see Gersten [18]). Platonov [50] proved that it is equivalent to the function $\exp^{(\log 2n)}(1)$, where $\exp^{(m)}(x)$ is the function defined by $\exp^{(0)}(n) = n$ and $\exp^{(k+1)}(n) = \exp(\exp^{(k)}(n)))$. However, $G_{(1,2)}$ is not residually finite (and in fact has very few finite quotients) [2]. Furthermore, the word problem in $G_{(1,2)}$ is decidable in polynomial time [41].

1.1.2 Method 2. Using subgroups with very large distortion

Consider a finitely presented group G and a "badly" distorted finitely generated subgroup H. Let $T = \langle G, t \mid t^{-1}ht = h\ (h \in H)\rangle$ be the HNN extension of G where the free letter t centralizes H. It was noticed by Bridson and Häfliger [11, Theorem 6.20.III] that the Dehn function of T is at least as large as the distortion function of H in G. The following result puts some limitations on this method of constructing complicated residually finite groups.

Lemma 1.1 *If the group T is residually finite then H is closed in the pro-finite topology of G.*

Proof In the notation above, suppose T is residually finite and $u \in G\backslash H$. Then $w = [u, t] \neq 1$ in T (by the standard properties of HNN extensions). Hence there exists a homomorphism ϕ from T onto a finite group T_w such that $\phi(w) \neq 1$. But this implies $\phi(u) \notin \phi(H)$ (since every element of $\phi(H)$ commutes with $\phi(t)$). Hence there exists a normal subgroup of finite index $N < G$ such that u does not belong to NH. In other words H must be closed in the pro-finite topology of G. □

Consider the following typical examples of residually finite groups G with highly distorted subgroups H. The first one is Wise's version [64] of Rips' construction [53] which for every finitely presented group Q gives a finitely presented residually finite small cancellation group G with a short exact sequence

$$1 \to N \to G \to Q \to 1$$

where N finitely generated. It is easy to see that the distortion function of N in G is at least as large as the Dehn function in Q, so choosing Q properly one can get a finitely presented residually finite group G with a highly distorted subgroup N. Now, the subgroup N is normal in the HNN extension T. So it is closed in the pro-finite topology of T only if $Q = G/N$ is residually finite. By Lemma 1.1, T can be residually finite only if Q is residually finite. In other words to construct a complicated finitely presented residually finite group T one has to

have the initial group Q complicated, finitely presented and residually finite as well.

The second example is the standard Mikhailova construction. In this case highly distorted subgroups of the direct product of two free groups $F_2 \times F_2$ can be obtained as equalizers of two homomorphisms $\phi_1 : E_1 \to M$ and $\phi_2 : E_2 \to M$ where E_1, E_2 are finitely generated subgroups of F_2 and M is finitely presented (see Sect. 5.2 below). But by Lemma 5.3 below the equalizer is closed in the pro-finite topology only if M is residually finite. Thus as in the previous example, in order to construct an algorithmically complex residually finite finitely presented group using Mikhailova's construction and the HNN extensions as above, one needs to have already a finitely presented residually finite algorithmically complex group M.

The third example is Cohen's [12] construction of highly distorted subgroups employing the modular machines. One can also prove that in that construction the subgroup will be pro-finitely closed only if the modular machine is very easy.

One can also try to use the hydra groups [10,14] to construct HNN extensions as above with Dehn functions bigger than any prescribed Ackermann function. The question of whether these groups are residually finite was open when the first version of this paper was written, and is now answered in negative in [49].

1.1.3 Method 3. Boone–Novikov constructions

One of the standard ways to produce algorithmically complicated groups is by simulating Turing machines using free constructions (HNN extensions and amalgamated products) which goes back to the seminal papers by Boone and Novikov (see, for example, [54]). There are currently many versions of that construction (for a recent survey see [57]). But in fact, it can be shown that for each known version of the proof of Boone–Novikov theorem using free constructions, even for easy Turing machines the corresponding group is non-residually finite. This is, for instance, the main idea of the example in [31]. Here is an even easier example. Let $G = G(M)$ be a group constructed by any of these constructions. Then for every input word u of the Turing machine M there exists a word $w = w(u)$ obtained by inserting some copies of u in $w(\emptyset)$ (here $\emptyset$ denotes the empty word), so that u is accepted by M if and only if $w(u) = 1$ in G. Now consider M that accepts a word a^n if and only if $n \neq 0$ (that machine is actually one of the basic building blocks in [59]). Then $w(a^n) = 1$ in G if and only if $n \neq 0$. Suppose that there exists a homomorphism ϕ onto a finite group H that separates $w(\emptyset) = w(a^0)$ from 1. Then $\phi(a)$ has finite order, say, s, in H. Therefore $\phi(w(\emptyset)) = \phi(w(a^0)) = \phi(w(a^s)) = \phi(1) = 1$, a contradiction. Hence $G(M)$ is not residually finite.

1.1.4 Method 4. Residually finite groups obtained by free constructions

In general, the question about residual finiteness of free constructions is very difficult. Currently there are only two large classes of groups where the question was settled: these are ascending HNN extensions of free groups [6] and certain groups acting "nicely" on CAT(0)-cubical complexes including small cancelation groups (see the recent work of Wise, for example, [25] and references therein). All these groups have

easy word problem and uniformly bounded Dehn functions. The reason for the lack of more examples is that groups obtained by free constructions from "nice" groups contain a lot of extra elements and it is not at all clear how to separate these elements from 1 by homomorphisms onto finite groups. In the two cases when it could be done, it was possible to reformulate the problem in the language of algebraic geometry and geometric topology, respectively.

1.1.5 Method 5. Other groups with complicated word problem

There are several other constructions of groups which simulate various algorithmic problems including undecidable ones, but each of these also always either produce non-residually finite groups or groups with simple word problem. For example, the group in [37] is based on the R. Thompson group V which is infinite and simple (hence not residually finite).

1.1.6 The main results of the paper.

In this paper, we construct finitely presented residually finite semigroups and groups with arbitrarily complex word problem, and also easy word problem but arbitrarily large (of course recursive) Dehn function, and arbitrarily large (recursive) depth function. We also give applications of these results to the questions about decidability of the universal theory of finite solvable groups and to distortion of pro-finitely closed subgroups of residually finite groups.

1.1.7 What next?

We expect the approach used in this paper to be useful in solving other problems that are still open. For example, the residually finite version of the Higman embedding theorem [24] would be very desirable. It is known [15,21,38] that a finitely generated recursively presented residually finite group may have undecidable word problem, and hence cannot be embedded into a finitely presented residually finite group. But it is not known whether undecidability of the word problem is the only obstacle for such an embedding. Thus it would be very interesting to find out whether every finitely generated residually finite group with decidable word problem embeds into a finitely presented residually finite group. Note that usually a version of Boone–Novikov theorem precedes a version of Higman theorem, hence we can consider this paper as a step toward the residually finite version of Higman's theorem.

1.2 The "yes" and "no" parts of the McKinsey algorithm

One of the initial motivations for studying residually finite groups, semigroups and other algebraic structures was McKinsey's algorithm solving the word problem in finitely presented residually finite algebraic structures.

Let $G = \langle X \mid R \rangle$ be a residually finite finitely presented algebraic structure of finite signature Σ (say, groups, semigroups, rings, etc.) Let us recall McKinsey's algorithm

solving the word problem in G (see [36,40]). Let $F(X)$ be the free algebraic structure of signature Σ freely generated by X. To solve the word problem in G one runs in parallel two separate algorithms $\mathcal{A}_{\text{yes}}$ and $\mathcal{A}_{\text{no}}$, such that starting with a given pair of elements $w, w' \in F(X)$ $\mathcal{A}_{\text{yes}}$ stops if and only if $w = W'$ in G and $\mathcal{A}_{\text{no}}$ stops if and only if $w \neq w'$ in G.

The algorithm $\mathcal{A}_{\text{yes}}$ enumerates one by one all consequences of the defining relations R and waits until $w = w'$ appears in the list.

The algorithm $\mathcal{A}_{\text{no}}$ enumerates all homomorphisms $\phi_1, \phi_2, \ldots,$ of G into finite algebraic structures of signature Σ and waits until $\phi_i(w) \neq \phi_i(w')$.

Note that although $\mathcal{A}_{\text{yes}}$ and $\mathcal{A}_{\text{no}}$ enumerate complimentary sets of elements of $F(X) \times F(X)$, these algorithms are quite different and their complexity functions can be very different too.

To estimate the complexity of algorithms and algorithmic problems, recall the basic definitions of the complexity theory. Let S be a set of objects (words, numbers, etc.) equipped with a size function $S \to \mathbb{N}$ so that there are only finitely many objects in S of any given size. For every algorithm $\mathcal{A}$ computing some partial function $\phi\colon S \to \mathbb{N}$, by $T_{\mathcal{A}}$ we denote the time function of $\mathcal{A}$ which is defined as the maximal running time of $\mathcal{A}$ on inputs of size at most n from the domain of ϕ (thus we simply ignore inputs not from the domain of ϕ). The time complexity of an algorithmic problem, i.e., the membership problem in a subset $X \subseteq S$ is the minimum of $T_{\mathcal{A}}$ for all algorithms $\mathcal{A}$ solving the problem, i.e., computing the indicator function of X. It is not that easy to define what "minimum of a set of functions" means. But we can define a pseudo-order on the set of functions as follows. For two functions f, g we write $f \preccurlyeq g$ (f is smaller than g) is there exists a constant C such that for any n, $f(n) \leq Cg(Cn) + Cn + C$. (The functions are equivalent if $f \preccurlyeq g$ and $g \preccurlyeq f$.) We do not need to address the question whether every set of functions has a minimum with respect to this order. We can certainly say whether the time complexity is polynomial (exponential, etc.).

Let now G be a finitely presented residually finite group. Although it is universally assumed that $\mathcal{A}_{\text{yes}}$ and $\mathcal{A}_{\text{no}}$ are very slow in general, there were no examples of groups G for which these algorithms were actually very slow.

Moreover it was not known if for some universal recursive function $f(n)$ the time complexity of the word problem in any finitely presented residually finite group (or semigroup) does not exceed $f(n)$.

In the case of finitely presented linear groups it is well known that the word problem can be solved in deterministic polynomial time [34,61]. This applies to most finitely presented groups (where "most" means "with overwhelming probability" in one of several probabilistic models): recent results of Agol [1] and Ollivier and Wise [45] together with the older result of Olshanskii [46] imply that most finitely presented groups are linear (even over $\mathbb{Z}$).

One of our main results is the following theorem (an immediate corollary of Theorem 4.21 below):

Theorem *For every recursive set of natural numbers X there exists a finitely presented residually finite solvable group G of class 3 such that the word problem in G is as hard as the membership problem in X.*

Remark 1.2 Note that if we replace "finitely presented" assumption by "recursively presented", then residually finite groups are known to be very complicated. As we have mentioned before, recursively presented finitely generated residually finite groups may have undecidable word problem [15,21,38]. See also the survey [16] where it is shown how to construct complicated residually finite groups using the method of Golod–Shafarevich.

Note also that although our groups are not linear, they are (Abelian of prime exponent)-by-linear since they are solvable of class 3 with the second derived subgroup Abelian of prime exponent (one can assume that the exponent is equal to any prime number, say, 2).

1.3 The time function of the algorithm $\mathcal{A}_{yes}$: the Dehn function

In [39] Madlener and Otto constructed finitely presented groups with arbitrarily large Dehn functions. For residually finite groups, the situation is different. Nilpotent groups are examples of residually finite groups with arbitrarily high polynomial Dehn function [4]. The Baumslag–Solitar groups $\langle x, y \mid x^y = x^k \rangle$, $k \geq 2$, are examples of residually finite (even linear) groups with exponential Dehn function. No examples of residually finite groups with bigger Dehn functions were known. This gap is filled by the following theorem (see Theorem 4.21).

For every recursive function $g(n)$ there exists a finitely presented residually finite solvable group G of class 3 such that the Dehn function of G is bigger than $g(n)$.

In addition we can assume that the time complexity of the word problem is polynomial.

1.4 The time function of the algorithm $\mathcal{A}_{no}$: the depth function

The time function of the algorithm $\mathcal{A}_{\text{no}}$ can be estimated in terms of the *depth function* introduced by Bou-Rabee [9]. Recall that if $G = \langle X \rangle$ is a finitely generated residually finite group or semigroup, the depth function $\rho_G(n)$ is the smallest function such that every two words $w \neq_G w'$ of length at most n are separated by a homomorphism to a finite group (semigroup) H with $|H| \leq \rho_G(n)$. That function does not depend on the choice of finite generating set X (up to the natural equivalence).

It is easy to see that for every finitely generated linear group or semigroup G, ρ_G is at most polynomial. Since finitely generated metabelian groups are subgroups of direct products of linear groups [62] the depth function of every finitely generated metabelian group is at most polynomial. By the recent result of Agol [1] based on the earlier results of Wise [63], every small cancelation group is a subgroup of a Right Angled Artin group, hence linear and has polynomial depth function. In fact one can have much smaller bounds for many linear groups. For example, for the free group F_2, $\rho_{F_2}(n) \preccurlyeq n^3$ by [9]. A lower bound for the depth function for a free group is equivalent to $n^{\frac{2}{3}}$ by a result of Kassabov and Matucci [26]. There are some finitely presented groups for which the depth function is unknown and very interesting. For example the ascending HNN extensions of free groups are known to be residually finite and even

virtually residually nilpotent (proved by Borisov and the third author [6,7]) but the only upper bound one can deduce from the proof is exponential. Although many of these groups have small cancelation presentations and so are covered by the results from [1], there are some groups of this kind for which the depth function is not known.

For finitely generated infinitely presented groups (even amenable ones) the situation is much clearer. Using the method of Kassabov and Nikolov [27] and the result of Nikolov and Segal [44] one can construct a finitely generated residually finite group with arbitrary large recursive depth function.

In this paper, we show that a similar result holds for finitely presented solvable of class 3 groups (see Theorem 4.22).

For every recursive function f one can construct a residually finite finitely presented solvable of class 3 *group with depth function greater than f.*

In addition, again, one can assume that the time complexity of the word problem in G is polynomial.

1.5 Methods of proof

As we have shown above (see Sect. 1.1.3) all versions of the Boone–Novikov construction ([12,37,54,59]) do not produce complicated residually finite groups. Instead, we simulate Minsky machines. We first simulate Minsky machines in semigroups and then "embed" these semigroups into solvable groups of class 3. The first simulation of Minsky machines in semigroups were studied by Gurevich [23] (see also [55,58]). Simulations of Minsky machines in groups, including solvable groups could be found in [60] and [28] (see also [32]). In this paper we closely follow the construction by the first author from her unpublished thesis [29].

Of course that construction also often leads to non-residually finite groups. But it turned out that the difficulty can be overcome by modifying the Minsky machine first. In this paper, we use the fact that every Turing machine recognizing a recursive set is equivalent to a *sym-universally halting* Minsky machine, i.e., Minsky machine whose symmetrization halts on every non-accepted configuration (see Theorem 2.7).

1.6 Structure of the paper

The paper is organized as follows. Section 2 contains preliminary results about Turing and Minsky machines that are needed further.

In Sect. 3, we simulate sym-universally halting Minsky machines in residually finite finitely presented semigroups and prove the analogs of the above theorems for semigroups. The advantage of using Minsky machines instead of the general Turing machines is that the resulting semigroups are much "smaller", and all non-zero elements of these semigroups are basically subwords of words corresponding to configurations of the Minsky machines. In the construction, we use the ideas from the papers of the third author [55,58] who proved that these semigroups are minimal in the following sense: the varieties generated by these semigroups are minimal varieties of semigroups containing finitely presented semigroups with undecidable word problem

(this leads to a complete description of varieties with decidable word problem where periodic groups are locally finite [55]).

In Sect. 4 we use semigroups from Sect. 3 to construct the groups with large Dehn and depth functions. The idea of such a construction came from the paper [28] of the first author solving a problem formulated by Adyan [33] by constructing a finitely presented group with undecidable word problem and satisfying a non-trivial identity. We use the simplification of that construction from the unpublished dissertation of the first author [29]. It turned out that if we start with residually finite semigroups from Sect. 3, we often get residually finite finitely presented groups whose Dehn and depth functions resemble the corresponding functions of the semigroups we start with. Finally, Sect. 5 contains two applications of our main results and methods. In Sect. 5.1, we strengthen the well-known result of Slobodskoi about undecidability of the universal theory of finite groups by showing that the universal theory of any set of finite groups that contains all finite solvable groups of class 3 is undecidable (see Theorem 5.1). This gives the first proper variety of groups where the set of all finite groups has undecidable universal theory. In Sect. 5.2, we construct pro-finitely closed subgroups of the direct product of two free groups with arbitrary large time complexity of the membership problem, distortion function and ther relative depth function (defined there).

2 Turing machines and Minsky machines

2.1 Turing machines

In this paper, we shall consider several types of machines. A machine M in general has an alphabet and a set of words in that alphabet called configurations. It also has a finite set of commands. Each command is a partial injective transformation of the set of configurations. A machine is called *deterministic* if the domains of its commands are disjoint. A machine usually has a distinguished configuration s_{acc} called the accept configuration and a set $I = I(M)$ of *input* configurations which recursively enumerate words in some finite alphabet A.

A machine halts if none of the commands is applicable to the configuration. We always assume that a machine halts on every accept configuration but it may halt on non-accept configurations too.

A *computation* Θ of M is a finite or infinite sequence of configurations and commands from P:

$$w_1 \xrightarrow{\theta_1} w_2 \xrightarrow{\theta_2} \dots \xrightarrow{\theta_l} w_{l+1}, \dots$$

such that $\theta_i(w_i) = w_{i+1}$ for every $i = 1, \dots, l, \dots$.

If the computation Θ is finite and w_{l+1} is the last configuration, then l is called the *length* of the computation, and we say that w_{l+1} is obtained from w_1 by applying Θ. A configuration is called *accepted* by M if there exists a computation connecting that configuration with s_{acc} (that computation is called *accepting*). The *time function*

$T_M(n)$ of M is the minimal function such that every accepted configuration of length $\leq n$ has an accepting computation of length $\leq T_M(n)$.

Let us give a definition of a Turing machine (see [59]). A Turing machine M with k tapes consists of hardware (the *tape alphabet* $A = \sqcup_{i=1}^{k} A_i$, and the *state alphabet* $Q = \sqcup_{i=1}^{K} Q_i$)[1] and program P (a list of commands, defined below). A *configuration* of a Turing machine M is a word

$$\alpha_1 u_1 q_1 v_1 \omega_1 \, \alpha_2 u_2 q_2 v_2 \omega_2 \, \cdots \, \alpha_K u_K q_K v_K \omega_K$$

(we included spaces to make the word more readable) where u_i, v_i are words in A_i, $q_i \in Q_i$ and α_i, ω_i are special symbols (not from $A \cup Q$). This machine has k *tapes*. For every configuration c the *content* of tape number i is the subword $\alpha_i u_i q_i v_i \omega_i$.

A command simultaneously replaces subwords $a_i q_i b_i$ by words $a_i' q_i' b_i'$ where $q_i, q_i' \in Q_i$, a_i, a_i', are either letters from $A_i \cup \{\alpha_i\}$ or empty, b_i, b_i' are either letters from $A_i \cup \{\omega_i\}$ or empty. A command cannot insert or erase α_i or ω_i, so if, say, $a_i = \alpha_i$, then $a_i' = \alpha_i$.

Note that with every command θ one can consider the *inverse* command θ^{-1} which undoes what θ does.

For the Turing machine we choose *stop states* q_i^0 in each Q_i, then a configuration w is accepted if there exists a computation starting with w and ending with a configuration where all state symbols are q_i^0 and all tapes are empty (which is the accept configuration for the Turing machine). Thus for a k-tape Turing machine s_{acc} is equal to $\alpha_1 q_1^0 \omega_1 \cdots \alpha_k q_k^0 \omega_k$.

Also we choose *start states* q_i^1 in each Q_i. Then an input configuration corresponding to a word u over A_1 is a configuration $\text{inp}(u)$ of the form

$$\alpha_1 u q_1^1 \omega_1 \, \alpha_2 q_2^1 \omega_2 \, \cdots \, \alpha_K q_K^1 \omega_K.$$

Thus the input set I consists of all these words. We say that a word u over A_1 is accepted by M if the configuration $\text{inp}(u)$ is accepted. The set of all words accepted by M is called the *language accepted by* M.

For every machine M, the machine $\text{Sym}(M)$ is made from M by adding the inverses of all commands of M. Two configurations w, w' are called *equivalent*, written $w \equiv_M w'$, if there exists a computation of $\text{Sym}(M)$ connecting these configurations. Clearly, $\equiv_M$ is an equivalence relation. For example, every two accepted input configurations are equivalent, because both are equivalent to s_{acc}.

The following general lemma is easy but useful.

Lemma 2.1 *Suppose that M is deterministic. Then*

(a) *Any reduced computation of* $\text{Sym}(M)$ *is a concatenation of two* (*possibly empty*) *parts $\Theta_1 \Theta_2^{-1}$, both Θ_1 and Θ_2 uses only commands of M.*
(b) *Two configurations w, w' of M are equivalent if and only if there exist two computations of M connecting w, w' with the same configuration w'' of M.*

[1] $\sqcup$ denotes disjoint union.

Proof Part (a) of the lemma follows from the fact that in any reduced computation of $\mathrm{Sym}(M)$ inverses of commands of M cannot be followed by commands of M (since M is deterministic).

For Part (b), if w is equivalent to w', then by Part (a) there is a computation of the form $\Theta_1\Theta_2^{-1}$ connecting w and w'. Then applying Θ_1 to w, and Θ_2 to w' produces the same configuration w''. □

Definition 2.2 (a) We say that an algorithmic problem A is *as hard as* an algorithmic problem B if for any decision algorithm for A which solves the problem in time T_A there exists an algorithm for B that solves it in time $\preccurlyeq T_A$.

(b) We say that a language Y *polynomially reduces* to a language X if there exists a deterministic Turing machine C checking membership in Y which uses an oracle checking membership in X and runs in polynomial time (in terms of the length of the word being checked.

(c) We say that languages X and Y are *polynomially equivalent* if there are polynomial reductions of X to Y and vice versa.

(d) We say that a machine M' *polynomially reduces* (resp. is *polynomially equivalent*) to a machine M if the configuration equivalence problem of M' polynomially reduces (resp. is polynomiall equivalent) to the configuration equivalence problem for M.

2.2 Universally halting Turing machines

A deterministic machine M is called *universally halting* if it does not have infinitely long computations (see [8]). We say that a computation of $Sym(M)$ is *reduced* if no command is followed by its inverse. We call a deterministic machine M *sym-universally halting* if $\mathrm{Sym}(M)$ does not have infinitely long reduced computations that start at a non-accepted configuration. It is proved in [13] that for every recursive set X of natural numbers, that is accepted by a deterministic Turing machine M there exists a universally halting deterministic Turing machine M' with one tape accepting X. From the construction, it is clear that M' polynomially reducible to M.

One can also convert a sym-universally halting Turing machine into a 1-tape sym-universally halting Turing machine:

Lemma 2.3 *Let M be a deterministic sym-universally halting Turing machine recognizing a language X. Then there exists a one-tape deterministic sym-universally halting Turing machine M' recognizing X and polynomially equivalent to M. Moreover there exists a map ϕ from the set $C(M)$ of configurations of M to the set $C(M')$ of configurations of M' such that*

(1) *For every word u in the alphabet of X, $\phi(\mathrm{inp}_M(u)) = \mathrm{inp}_{M'}(u)$*
(2) *For every $c \in C(M)$, $|\phi(c)| = O(|c|)$*
(3) *$\phi(c) \equiv_{M'} \phi(c')$ if and only if $c \equiv_M c'$*
(4) *If a command θ of M takes c to c', then there exists a computation of M' of length at most $O(|c|)$ that takes $\phi(c)$ to $\phi(c')$.*

Proof The proof is basically by inspection of the proof from [48, Theorem 2.1]. Recall the way to convert a k-tape Turing machine M into a 1-tape Turing machine M'. The tape alphabet of M' consists of all letters occurring in the configurations of M. A configuration of M' is a word $\alpha c_1 q c_2 \omega$ where $c = c_1 c_2$ is a configuration of M or differs from a configuration of M by at most two letters: the left and right neighbor of some state letter. The map ϕ takes each configuration c to which a command of M' is applicable to the configuration $\alpha c q_\theta \omega$ where q_θ is a state letter of M' corresponding to the command θ that is applicable to c (note that since M is deterministic, θ is determined by c). If no command of M is applicable to c, we set $\phi(c) = \alpha c q \omega$ where q is a distinuished state letter of M'.

A command θ of M that substitutes $a_i q_i b_i$ by $a'_i q'_i b'_i$, $i = 1, \ldots, k$, is simulated as follows: the letter q_θ moves from right to left, and every time it meets q_i, it checks if it is a part of the subword $a_i q_i b_i$, and if so, replaces it by $a'_i q'_i b'_i$. After all these substitutions the letter q_θ returns to the right end of the configuration (next to the letter α) and becomes ready to simulate the next command of M or becomes the distinguished state letter q (for more details see [48]). It is easy to check (using Lemma 2.1) that the new machine is polynomially equivalent to the old one and properties (1)–(4) hold. It is also easy to check that if the original machine is sym-universally halting, the new one is also sym-universally halting. □

Theorem 2.4 *For every recursive language X there exists a deterministic sym-universally halting Turing machine M with one tape recognizing X. Moreover if M' is any deterministic Turing machine recognizing X then we can additionally assume that M polynomially reduces to M'.*

Proof Let M' be a deterministic universally halting Turing machine with k tapes recognizing X. Consider the new Turing machine M'' constructed as follows. M'' has one more tape than M', called the *history* tape. The alphabet A'' of this tape is in one-to-one correspondence with the set of commands P of M: $A'' = \{[\theta], \theta \in P\}$. An input configuration of M'' does not have letters from A'' and its subword written on the first k tapes is an input configuration of M'. With every command θ' of M' we associate a command θ'' of M''. It does what θ' would do on the first k tapes of M'' and inserts $[\theta]$ on the history tape of M'' next to the right of q_{k+1}. After the first k tapes of M'' form the accept configuration $s_{\mathrm{acc}}(M')$, the machine erases letters from the tape alphabet A'' on the history tape and halts, producing the accept configuration $s_{\mathrm{acc}}(M'')$ of M'' (thus $s_{\mathrm{acc}}(M'') = s_0 \alpha_{k+1} q^0_{k+1} \omega_{k+1}$).

Let P'' be the program of M''. We shall modify M'' further to obtain a new $(k+1)$-tape Turing machine M. It has the same tape alphabets as M'' and all state letters of M'' are also state letters of M. The input and accept configurations are also the same. The program P of M contains a copy $\tilde{P}$ of P'' (the set of the *main commands*) and some new commands. After each main command $\tilde{\theta}$ of $\tilde{P}$ which does the same as the corresponding command θ of M'', but changes the state letters to state letters which are not in M'', M executes the history written on the history tape backward, without erasing the history tape. It just scans the history tape from left to right, reading the symbols written there one by one and executing on the first k tapes the inverses of the commands written on the history tape. If at the end of the scanning the history tape, the word written on the first k tapes is an input configuration of M'', M executes on

the first k tapes the history written on the history tape in the natural order, scanning the history tape from right to left. After that M changes the state letters to what θ would do, and is ready to execute the next main command. We do not give precise definition of the program of M because it is obvious on the one hand and long on the other hand. The machine M is deterministic and universally halting. Moreover for every input configuration c of M'', M accepts c if and only if c is accepted by M'', hence if and only if c corresponds to a word from X.

Note that since M is deterministic, and no commands of it are applicable to the accept state, M accepts the same language as M'.

Let c be a configuration of M that is not accepted by M. By Lemma 2.1(a) every reduced computation of $\mathrm{Sym}(M)$ starting at c is a concatenation of a computation of M followed by a computation of the machine M^{-1} obtained from M by replacing every command with its inverse. Since M is universally halting, there are only finitely many computations of M starting with c.

Claim 1 There are finitely many computations of M^{-1} starting with any non-accepted configuration c. Equivalently, there are only finitely many computations of M ending with c.

Indeed, since c is not accepted, none of the tape letters on the history tape is erased during any computation ending in c. Therefore by the definition of M every computation Θ ending at c and having length $\geq |c|$ must arrive at a configuration $c = c_1c_2$ where c_1 is an input configuration of M' and c'' is the content of the history tape of c with the state letter moved next to ω_{k+1}. And, moreover, this should happen at most $|c|$ steps before arriving to c. The suffix c_2 of c' is completely determined by c. The sequence of commands from Θ used to get from c' to c is in one-to-one correspondence with the sequence of tape letters of c_2 to the right of the state letter. Therefore c' is completely determined by c. If the length of the computation Θ is at least $2|c|$, then a configuration of the form c_1c_2' must occur in it before c', where c_2' differs from c_2 only by the state letter (which, as in c_2 is next to ω_{k+1}. This is impossible because between every two arrivals at such configurations, every computation of M'' must execute one of the main commands, and increase the number of tape letters on the history tape. Thus we proved

Claim 2 Every computation ending at c is of length less than $2|c|$.

Claim 2 implies Claim 1 and the fact that M is sym-universally halting.

To prove that M polynomially reduces to M', let c, c' be two configurations of M. In order to check whether c and c' are equivalent first check whether c is accepted. For this we need to run the program of M for at most $2|c|$ steps and see whether we first get a configuration c' of the form c_1c_2 where c_1 is an input configuration of M' and c_2 is of the form $\alpha_{k+1}vq_{k+1}\omega_{k+1}$ ($q_{k+1} \in Q_{k+1}$) and then a configuration of the form $c_1\alpha q'_{k+1}v\omega_{k+1}$. If so, then check the equivalence $c_1 \equiv_{M'} s_{\mathrm{acc}}(M')$ using the oracle that checks equivalence of configurations of M'. The answer is "yes" if and only if c is accepted. That process takes linear (in $|c|$) number of steps and one oracle query. Similarly, we check if c' is accepted. If Both c and c' are accepted, then $c \equiv_M c'$. If one of them is accepted and another one is not, then these configurations are not equivalent.

Finally suppose that both c and c' are not accepted. By Lemma 2.1, $c \equiv_M c'$ if and only if there exist two computations Θ_1 starting with c_1 and Θ_2 starting with c_2 such that the end configurations of these computations are the same. We can assume that either Θ_1 or Θ_2 has length $>2(|c| + |c'|)$ (that can be checked in time linear in $|c| + |c'|$). Without loss of generality assume that the length of Θ_1 is bigger than $2|c|$. Then after at most $2|c|$ steps of Θ_1 we arrive at a configuration of the form c_1c_2 where c_1 is a configuration of c' such that there exists a computation Θ' of length $<|c_2|$ of M' starting with an input configuration d of M' and ending at c_1, and the sequence of commands used in this computation is in one-to-one correspondence with the tape letters in c_2. Therefore c is equivalent to a configuration of the form $d\alpha_{k+1}q_{k+1}\omega_{k+1}$ where $q_{k+1} \in Q_{k+1}$ which is an input configuration of M (and we need linear time in terms of $|c|$ to find this configuration). Hence we can assume that $c = d\alpha_{k+1}q_{k+1}\omega_{k+1}$. If the longest computation of M starting at c' has length $<2|c'|$, then the longest configurations we can reach by one of these computations is at most $2|c'|$, and it would take at most $O((|c'| + |c|)^2)$ steps of M to reach any of these configurations starting at c. Thus it would take polynomial time to check whether $c \equiv_M c'$ in this case. Thus we can assume that there exists a computation of M that starts at c' and has length $>2|c'|$. Then, as before we can assume that $c' = d'\alpha_{k+1}q_{k+1}\omega_{k+1}$, where d' is an input configuration of M. Now if order to check if $c \equiv_M c'$ it is enough to check whether $d \equiv_{M'} d'$ which is one oracle query.

It remain to apply Lemma 2.3 and convert M to a 1-tape Turing machine. □

2.3 Minsky machines

The hardware of a k-glass Minsky machine MM_k, $k \geq 2$, consists of k glasses containing coins. We assume that these glasses are of infinite height. The machine can add a coin to a glass, and remove a coin from a glass (provided the glass is not empty).

The commands of a Minsky machine are numbered started at 0. A configuration of a k-glass Minsky machine is a $k + 1$-tuple $(i; \epsilon_1, \ldots, \epsilon_K)$ where i is the number of the command that is to be executed, ϵ_j is the number of coins in the glass $\#j$. We can write a number in the unary notation: the number n is written as $1 \ldots 1$ (n ones). Clearly then we can view a configuration as a word in the alphabet consisting of digits 1 and symbols (,), ; and comma ",". The accept configuration is $s_0 = (0; 0, \ldots, 0)$ (the command number is 0, all glasses are empty) and input configurations have the form $(1; m, 0, \ldots, 0)$.

Let us describe commands of Minsky machines more precisely. Each basic command has one of the following forms:

- Put a coin in each of the glasses $\#\#n_1, \ldots, n_l$ and go to command $\# j$. We shall encode this command as

$$i; \rightarrow \text{Add}(n_1, \ldots, n_l); j$$

  where i is the number of the command;
- Provided the glasses $\#\#n_1, \ldots, n_l$ are not empty, take a coin from each of these glasses and go to instruction $\# j$. This command is encoded as

$$i; \epsilon_{n_1} > 0, \ldots, \epsilon_{n_l} > 0 \to Sub(n_1, \ldots, n_l); j;$$

- Provided glasses $\#\#n_1, \ldots, n_l$ are empty, go to instruction $\# j$. This command is encoded as

$$i; \epsilon_{n_1} = 0, \ldots, \epsilon_{n_l} = 0 \to j;$$

- Stop. This command is encoded as $i; \to 0;$

Remark 2.5 This defines deterministic Minsky machines. We will also need non-deterministic Minsky machines. Those will have two or more commands with the same number.

Remark 2.6 We can also use (composite) commands that are not literally commands described above but can be easily split into a few basic commands, such as "Put coins in glasses $\#\#i_1, \ldots, i_l$ provided glasses $\#\#n_1, \ldots, n_m$ are empty".

Theorem 2.7 *Let X be a recursively enumerable set of natural numbers. Then the following holds:*

(a) *there exists a 2-glass deterministic Minsky machine MM_2 which "enumerates" X in the following sense: for every $m \in \mathbb{N}$, if $m \in X$, then MM_2 takes configuration $(1; 2^m, 0)$ to the accept configuration $(0; 0, 0)$, if $x \notin X$, then starting with $(1; 2^m, 0)$ it works infinitely long time.*

(b) *If X is recursive, then there exists a 2-glass deterministic Minsky machine MM_2 that recognizes X and is sym-universally halting.*

(c) *In (a) and (b) we can also assume for every computation of MM_2 starting with a configuration c and each of the glasses, that glass is emptied after at most $O(|c|)$ steps (here $|c|$ denotes the* size *of configuration c, i.e., the total number of coins in all glasses of the configuration).*

(d) *If M is a deterministic Turing machine recognizing X, then we can assume that MM_2 polynomially reduces to M.*

Proof The proof of Part (a) can be found in [35]. To prove (b) let us first recall the way to convert a 1-tape Turing machine M into a 2-glass Minsky machine MM_2 [35], and then prove that if M is sym-universally halting, then so is MM_2.

Suppose that the tape alphabet of M has m letters. For simpliciy consider the case when $m = 2$. The general case is absolutely similar. So suppose that the set of tape letters is $\{1, 2\}$. Let $q_1, \ldots, q_s$ be the set Q of state letters. With every configuration $c = \alpha u q_i v \omega$ of M we associate the following configuratiion of a 3-glass Minsky machine MM_3: $\phi(s) = (i; n_u, n_{\vec{v}}, 0)$ where n_u is the word u viewed as a natural number written in base 3, and $\vec{v}$ is the word v read from right to left. Now every command $aq_ib \to a'q_jb'$ of M is interpreted by MM_3 as follows. If a, a' are not empty, the machine MM_3 needs to check whether $n_u \equiv a \pmod 3$, $n_{\vec{v}} \equiv b \pmod 3$ and if so, then replace the last digit of n_u by a' and the last digit of $n_{\vec{v}}$ by b'. If, say, a is empty, then the machine should just multiply n_u by 3 and add a'. For example, the command θ of the form $1q_i2 \to q_j1$ is interpreted by a sequence $P(\theta)$ of commands of MM_3 as follows. The commands of $P(\theta)$ will be numbered $i.1$ through $i.l$ for some

l. In order to check that $n_u \equiv 1 \pmod 3$, the machine should remove one coin from the first glass, then repeatedly keep removing 3 coins from the first glass while adding 1 coin to the third glass. If at the end the first glass is empty, then, indeed, $n_u \equiv 1 \pmod 3$. If this is the case (otherwise the command is not applicable and the machine halts), we remove the coins from the third glass, and check if $n_{\vec{v}} \equiv 2 \pmod 3$. If so, we multiply the number of coins in the second glass by 3 (keep adding 3 coins to the second glass while removing a coin from the third glass until the third glass is empty), and then add two coins to the second glass.

It is easy to see that if c is a configuration of M, then after applying $P(\theta)$ to $\phi(c)$, we get $\phi(\theta(c))$. The length of the computation of $P(\theta)$ connecting $\phi(c)$ with $\phi(\theta(\phi))$ is $O(|c|)$.

Let MM_3 be the resulting 3-glass Minsky machine. In order to convert it into a 2-glass Minsky machine, we associate with every configuration $(c = i, m, n, p)$ of MM_3 the following configuration $\psi(c)$ of a 2-glass Minsky machine: $(i; 2^m 3^n 5^p, 0)$. Now removing (adding) a coin from (to) glass number j of MM_3 where $j = 1, 2$ or 3 is simulated by dividing (multiplying) the number of coins in the first glass by 2,3 or 5 respectively.

Notice that the following property of MM_3 holds:

(*) If Θ is a computation of MM_3 and $c_1, c_2, \ldots, c_m$ are configurations occurring in that computation, and $m > \ell|c_1|$ for some universal constant ℓ, then one of the configurations $c_2, \ldots, c_m$ must be of the form $\phi(c')$ where c' is a configuration of M. Moreover if $\phi(c')$ and $\phi(c'')$ are two consecutive configurations of that form in Θ, then there exists a command θ' of M such that $\theta'(c') = c''$.

Therefore if there exists an infinite computation of MM_3 or MM_3^{-1} starting with some configuration c of MM_3, then there exists an infinite computation of M (resp. M^{-1}) starting with some configuration c'. Moreover if c is not accepted by MM_3, then c' is not accepted by M. Thus if M is sym-universally halting, then MM_3 is sym-universally halting. The proof for MM_2 is similar. This gives Part (b).

Part (c) of the theorem immediately follows from the construction.

Part (d) is proved as follows. Suppose that c, c' are two configurations of MM_3 (for MM_2 the proof is similar). By (*) in at most $O(|w|)$ steps of MM_3 either c turns into a configuration of the form $\phi(d)$ for some configuration d of M or MM_3 halts. In the latter case, we check whether c is equivalent to c' in $O(|c|)$ steps. So we can assume that both c and c' are equivalent to configurations $\phi(d)$ and $\phi(d')$ for some configurations d, d' of M, and the lengths of d, d' are $O(|c| + |c'|)$. Again by (*), c is equivalent to c' if and only if d and d' are equivalent configurations of M. Thus we need to use the oracle once. □

3 Simulation of Minsky machines by semigroups

3.1 The construction

Here we will show how to simulate a Minsky machine by a semigroup. The construction is based on the following general idea which applies also in the case of solvable groups considered later.

First, with every configuration ψ we associate a word $w(\psi)$. Then with every command κ of the Minsky machine M we associate a finite set of defining relations R_κ. The semigroup $S(M)$ is defined by the relations from the union R of all R_κ (which is finite since we have only a finite number of commands) and usually some auxiliary relations Q which are in a sense "independent" of R but make the semigroup "smaller". We need Q, for example, to make sure $S(M)$ satisfies a particular identity.

We say that the semigroup $S(M)$ *simulates* M if the following holds for arbitrary configurations ψ_1, ψ_2 of M:

$$\psi_1 \equiv_M \psi_2 \text{ if and only if } w(\psi_1) = w(\psi_2) \text{ in } S(M). \tag{1}$$

Usually, in order to prove the property (1) one has to prove the following two properties.

Property 3.1 If a configuration c' can be obtained from a configuration c by a command κ of M then the word $w(c')$ can be obtained from the word $w(c)$ by applying defining relations of $S(M)$ from the set R_κ.

Property 3.2 If a word $w(c')$ can be obtained from a word $w(c)$ by applying the defining relations of $S(M)$ then $c \equiv_M c'$.

It is easy to see that Properties 3.1 and 3.2 imply property (1).

There is an easy way to interpret Minsky machines in a semigroup $S(M)$. Let MM_k be a Minsky machine with k glasses and commands ##1, 2, ..., N, 0 (here the command number 0 is the stop command, it is the command with domain $\{0; 0, \ldots, 0)\}$). Then $S(MM_k)$ is generated by the elements $q_0, \ldots, q_N$ and $\{a_i, A_i \mid i = 1, \ldots, k\}$. The set of defining relations of $S(MM_k)$ consists of all relations

$$a_i a_j = a_j a_i, a_i A_j = A_j a_i, A_i A_j = A_j A_i, i \neq j, \tag{2}$$

which we shall call *commutativity relations*, all relations of the form $xy = 0$ where xy is a two-letter word which is *not* a subword of a word of the form $q_i a_1^{\epsilon_1} \ldots a_k^{\epsilon_k} A_1 \ldots A_k$ modulo the commutativity relations (2) (for example $q_i q_j = A_i a_i = a_i q_j = A_i q_j = 0$), which we shall call 0-*relations*, and relations associated with commands of M according to the following table,

Command of MM_k	Relation of $S(MM_k)$
$i \to \text{Add}(n_1, \ldots, n_m); j$	$q_i = q_j a_{n_1} \ldots a_{n_m}$
$i, \epsilon_{n_1} > 0, \ldots, \epsilon_{n_m} > 0 \to \text{Sub}(n_1, \ldots, n_m); j$	$q_i a_{n_1} \ldots a_{n_m} = q_j$
$i, \epsilon_{n_1} = 0, \ldots, \epsilon_{n_m} = 0 \to j$	$q_i A_{n_1} \ldots A_{n_m} = q_j A_{n_1} \ldots A_{n_m}$

(3)

These will be called the Minsky relations.

The words in $S(MM_k)$ corresponding to configurations of M are the following:

$$w(i;\epsilon_1,\dots,\epsilon_k) = q_i a_1^{\epsilon_1}\dots a_k^{\epsilon_k} A_1\dots A_k.$$

The proof that Properties 3.1 and 3.2 hold in $S(MM_k)$ follows easily from Lemma 2.1, see [32,55].

3.2 Residually finite finitely presented semigroups

The following obvious lemma shows that the auxiliary 0-relations make the semigroup $S(MM_k)$ really small: it just does not have too many elements which are not related to configurations of MM_k.

Lemma 3.3 *Every word W in the generators of $S(MM_k)$ that is not equal to 0 in $S(MM_k)$ is, modulo the commutativity relations, a subword of some word of the form $w(i;\epsilon_1,\dots,\epsilon_k)$.*

Lemma 3.4 *Suppose that a word W is not 0 in $S(MM_k)$. By Lemma 3.3 W is a subword of a word of the form $w(i;\epsilon_1,\dots,\epsilon_k)$ (up to the commutativity relations). Suppose that W does not contain either q_i or one of the A_j. Then there are at most $O(|W|)$ different (up to the commutativity relations) words that are equal to W in $S(MM_k)$. All these words are subwords of words of the form $w(i';\epsilon_1,\dots,\epsilon_k)$ such that the configurations $(i;\epsilon_1,\dots,\epsilon_k)$ and $(i',\epsilon_1',\dots,\epsilon_k')$ of M are equivalent.*

Proof If W does not contain q_i, then the only relations that apply to W are the commutativity relations, so the only words that are equal to W in $S(M)$ are the words obtained from W by the use of commutativity relations.

Suppose that W contains q_i but does not contain one of the A_j.

Without loss of generality, we can assume that W contains every letter from $w(i;\epsilon_1,\dots,\epsilon_k)$ except some of the A_j's.

Every application of a Minsky relation to W corresponds to a command of the Minsky machine, applied to the configuration $c = (i;\epsilon_1,\dots,\epsilon_k)$. Let $c = c_1 \to c_2 \to \dots$ be any computation of $\mathrm{Sym}(MM_k)$ starting with c. Then the sequence of commands of MM_k applied in that computation has the form $\Theta_1\Theta_2^{-1}$ where Θ_1, Θ_2 are computations of MM_k (by Lemma 2.1). Each computation Θ_i corresponds to a sequence Y of applications of Minsky relations and commutativity relations (by Property 3.1). If that sequence of relations can be applied to W, then this computation never checks whether glass $\#j$ is empty. By Property (c) of Theorem 2.7, the lengths of Θ_1 and Θ_2 must be at most $O(|W|)$. This implies the statement of the lemma. □

Recall that s_0 is the accept configuration of MM_k, i.e., $s_0 = (0;0,\dots,0)$.

Lemma 3.5 *Suppose that the Minsky machine MM_k is sym-universally halting. Then*

(a) *Every element z of $S(MM_k)$ which is not equal to 0 or $w(s_0)$ has finitely many divisors, i.e., elements y such that $z = pyq$ for some $p, q \in S(MM_k) \cup \{1\}$.*
(b) *For every configuration c of MM_k the word $w(c)$ is equal to $w(s_0)$ in $S(MM_k)$ if and only if ψ is accepted by MM_k.*

Proof (a) If z is represented by a word w that contains one of the q_i and all letters A_j, then it must be equal to one a word of the form $w(i; \epsilon_1, \ldots, \epsilon_k)$ modulo commutativity relations (by Lemma 3.3). In that case applying relations of $S(MM_k)$ to w amounts to applying commands of $\mathrm{Sym}(MM_k)$ to the configuration $c = (i; \epsilon_1, \ldots, \epsilon_k)$ (by Properties 3.1 and 3.2). Thus every divisor of z is represented by a subword of one of the words $w(c')$ such that $c \equiv_{MM_k} c'$. Also note that c cannot be an accepted configuration of MM_k because otherwise z would be equal to $w(s_0)$ in $S(MM_k)$. Since MM_k is sym-universally halting, the number of configurations that are equivalent to c is finite. Hence the number of divisors of z is finite too.

If a word w representing z does not contain a q-letter or one of the A_i, then we can apply Lemma 3.4.

(b) This follows from Properties 3.1 and 3.2.

□

Remark 3.6 Note that for every element z of any semigroup S the set of all non-divisors of z in S is an ideal (denoted by $N(z)$). By definition of a divisor, $N(z)$ does not contain z. The Rees quotient semigroup $S/N(z)$ consists of all divisors of z and 0 with a natural multiplication. In particular, if z has only finitely many divisors then $S/N(z)$ is finite. Also note that for every ideal I of S, if $z \notin I$, then z is separated from every other element of S by the natural homomorphism from S to S/I.

Lemma 3.7 *If MM_k is sym-universally halting, then $S(MM_k)$ is residually finite.*

Proof Suppose that MM_k is sym-universally halting. Let $z_1 \neq z_2$ be two different elements of $S(MM_k)$. We need to show that there exists a homomorphism from $S(MM_k)$ to a finite semigroup separating z_1 and z_2.

First suppose that either z_1 or z_2 is not in $\{0, w(s_0)\}$. Then by Remark 3.6 z_1 and z_2 are separated by one of the natural homomorphisms from $S(MM_k)$ to $S(MM_k)/N(z_1)$ or $S(MM_k)/N(z_1)$ which are finite semigroups by Lemma 3.5 (a). Thus we can assume that $z_1 = 0, z_2 = w(s_0)$. Consider the (finite) set U of all subwords of the word $A_1 A_2 \ldots A_k$, including the empty word $\emptyset$. We identify words in U which are equal modulo the commutativity relations. For each $u \in U$ let us introduce a symbol κ_u. Now consider the finite set $L = \{0, \alpha_i, A_i, \kappa_u \mid i = 1, \ldots, k, u \in U\}$ with the following operation: $\kappa_u \alpha_i = \kappa_u$ if u does not contain the letter A_i otherwise $\kappa_u \alpha_i = 0$, $\kappa_u A_i = \kappa_{uA_i}$, if u does not contain A_i and $\kappa_u A_i = 0$ otherwise, $\alpha_i A_i = A_i, \alpha_i^2 = \alpha_i, \alpha a_i A_j = A_j \alpha_i, \alpha_i \alpha_j = \alpha_j \alpha_i, A_i A_j = A_j A_i$ for every $i \neq j$ between 1 and k, all other products are equal to 0. Then it is easy to see that L is a finite semigroup and the map $q_m \to \kappa, a_i \to \alpha_i, A_i \to A_i$ extends to a homomophism from $S(MM_k)$ to L separating z_1 and z_2. □

Recall that the *Dehn function of a finite semigroup presentation* $\langle X \mid R \rangle$ is the minimal function $f(n)$ such that for any words u, v which are equal in S and such that $|u| + |v| \leq n$, there exists a derivation of length at most $f(n)$ of this equality from the defining relations. For a finitely presented semigroup, a Dehn function does not depend on the choice of finite presentation (up to equivalence), and the equivalence class of that function is called the *Dehn function* of the semigroup.

Remark 3.8 The time complexity of the word problem is bounded from above in terms of the Dehn function of a finitely presented semigroup S: given the Dehn function $f(n)$ of a semigroup presentation $\mathcal{P}$, in order to check whether $w = w'$ (mod $\mathcal{P}$) with $|w| + |w'| \leq n$, we just need to check all sequences of length $\leq f(n)$ of applications of defining relations $w \to w_1 \to \ldots$. moreover the word problem is decidable if and only if the Dehn function is recursive [39].

Theorem 3.9 *For every recursive set of natural numbers X and every recursive function $g(n)$ there exists a finitely presented residually finite semigroup S such that the word problem in S is as hard as the membership problem in X and polynomially reduces to it; the Dehn function S is bigger than $g(n)$.*

Proof By Theorems 2.4 and 2.7 there exists a sym-universally halting 2-glass Minsky machine that recognizes X and whose configuration equivalence problem polynomially reduces to the membership problem in X. By Lemma 3.5, the problem of recognizing equality to $w(s_0)$ in $S(MM_2)$ is at least as hard as the membership problem in X. By Lemma 3.7, $S(MM_2)$ is residually finite. □

Remark 3.10 The proof of Theorem 3.9 could be simplified a little if instead of the semigroup $S(MM_k)$ we consider the semigroup $\tilde{S}(MM_k)$ obtained from $S(MM_k)$ by adding one relation $q_0 = 0$. That is $\tilde{S}(MM_k)$ is the Rees quotient of $S(MM_k)$ by the ideal generated by q_0. Indeed, if MM_k is sym-universally halting, then in $\tilde{S}(MM_k)$ every non-zero element has only finitely many divisors, and so it is residually finite by [22]. The word problem in $\tilde{S}(MM_k)$ and the word problem in $S(MM_k)$ are polynomially equivalent. That follows from the fact that no command of MM_k apply to a configuration of the form $(0; m, n)$. The semigroup $\tilde{S}(MM_k)$ is used in the next subsection.

3.3 Residually finite semigroups with large depth function

Recall the definition of the depth function ρ: for every finitely generated residually finite semigroup S and every number n, $\rho_S(n)$ is defined as the smallest number such that for every two different elements z, z' in S of word length $\leq n$ there exists a homomorphism ϕ from S onto a finite semigroup B of cardinality at most $\rho_S(n)$ such that $\phi(z) \neq \phi(z')$.

The following lemma from [22] follows from Remark 3.6.

Lemma 3.11 *Suppose that every non-zero element of a semigroup S with 0 has finitely many divisors. Then S is residually finite.*

Theorem 3.12 *For every recursive set of natural numbers X and every recursive function $g(n)$ there exists a finitely presented residually finite semigroup S such that the depth function of S is bigger than $g(n)$. In addition, the word problem in S and the membership problem in X polynomially reduce to each other.*

Proof Let MM_2 be a sym-universally halting 2-glass Minsky machine with $N + 1$ commands numbered $0, \ldots, N$. We need the following property of MM_2:

(**) No command of MM_2 makes the machine to go to the command number 1 (the start command).

Clearly that property can be achieved by renumbering commands starting from 2 (use numbers $0, 2, \ldots, N+1$ instead of $0, 1, \ldots, N$), and then adding command $1; \to 2$. The new Minsky machine accepts the same set of numbers and is still sym-universally halting with the same (up to the equivalence) time function. Thus we will assume that MM_2 satisfies (**).

We can also assume without loss of generality that the machine MM_2 can go to the stop command number 0 only if both glasses of it are empty. Indeed this can be done by replacing the stop command number 0 by the command

$$N+1; \epsilon_0 = 0, \epsilon_1 = 0 \to 0$$

and then adding the stop command number 0.

Consider the following new, non-deterministic Minsky machine MM_4. Its hardware consists of the 2 glasses of MM_2 plus two more glasses, numbers 3 and 4. In every command of MM_4 we add the instruction to add a coin to glass 3 provided glass 4 is empty. Thus if a command was $i; \to \text{Add}(1, 2); j$, we replace it by two commands $i; \epsilon_4 = 0 \to i', i'; \to \text{Add}(1, 2); j$ where i' is a new number (not the number of any other command of the Minsky machine).

These commands will be called *old*.

Also for every $i = 1, \ldots, N$ we add the following new command

$$i; \text{Add}(3, 4) \to i \tag{4}$$

and for every $i = 2, \ldots, N$ we add the following new command.

$$i; \epsilon_3 = 0, \epsilon_4 = 0 \to 0 \tag{5}$$

Thus we obtain a non-deterministic Minsky machine (see Remark 2.6) where there will be three commands numbered by each $i = 2, \ldots, N$: one old command and the two new ones, and two commands number 1. The new command (4) allows us to add, at any step of the computation, except the first one, equal (but arbitrary) number of coins in glasses 3 and 4, and if both glasses 3 and 4 are empty, the computation can stop. But we can execute an old command only when the glass 4 is empty, so a new command cannot be followed by a modified command of MM_2. Thus the following two remarks about MM_4 are straightforward.

Remark 3.13 (a) Glass 3 cannot be empty in any configuration of a computation of MM_4 except possibly the first one.

(b) In every reduced computation of $\text{Sym}(MM_4)$ a new command cannot be followed by an old command or its inverse.

If a computation of MM_4 starts at an input configuration $(1; m, 0, 0, 0)$, then by Remark 3.13(a) command (5) cannot be applied in that computation (that is why we did not include command $1; \epsilon_3 = 0, \epsilon_4 = 0 \to 0$ in the program of MM_4), and so the

computation can end with the configuration $s_0 = (0; 0, 0, 0, 0)$ if and only if $(1; m, 0)$ is accepted by MM_2. Therefore a number m is accepted by MM_4 if and only if m is accepted by MM_2.

Let us say that the commands coming from MM_2 have weight 1 and new commands (4), (5) have weight 0. The weight of a computation is then the sum of the weights of all commands used in the computation. We also define the weight of a configuration as the number of coins in the first 3 glasses minus the number of coins in glass 4. Every computation C of $\mathrm{Sym}(MM_4)$ projects onto a computation $\pi(C)$ of $\mathrm{Sym}(MM_2)$: we simply forget the extra two glasses and the new commands. The weight of C is equal to the length of $\pi(C)$. The numbers of coins used in C and $\pi(C)$ in each of the first 2 glasses are the same, the number of coins in glass 3 in the last configuration c of C minus the number of coins in glass 4 of c is equal to the weight of C.

The machine MM_4 is not deterministic, but some form of Lemma 2.1 still holds.

Remark 3.14 Every computation of $\mathrm{Sym}(MM_4)$ has the form $\Theta_1^{-1}\Theta_2\Theta_3^{-1}\Theta_4$ where each Θ_i is a computation of MM_4, all commands used in Θ_1 (resp. Θ_4) are new and have the same number, all commands used in Θ_2, Θ_3 are old. Indeed, it follows from Remark 3.13 (b) and the fact that MM_2 is deterministic (and so Lemma 2.1 applies to MM_2).

Also any computation C of MM_2 lifts to a set $\mathcal{C}$ of (possibly infinitely many) computations of MM_4, the weight of each $C' \in \mathcal{C}$ is the same as the length of C, and the number of coins used in each of the first 2 glasses is the same.

This implies that if MM_2 is sym-universally halting, then for every configuration c of MM_4 the weights of every computation C of $\mathrm{Sym}(MM_4)$ starting with c without repeated configurations are bounded, the number of coins in the first 2 glasses of MM_4 used during any of these computations is bounded, and the weights of configurations appearing in this computation are bounded (all these bounds depend only on c).

Consider the semigroup $\tilde{S}(MM_4)$ i.e., the quotient of $S(MM_4)$ by the ideal generated by q_0 (see Remark 3.10). Every non-zero element w in $\tilde{S}(MM_4)$ is represented (modulo the commutativity relations) by a word of the form $u(w)v(w)$ where

$$u(w) = q_i^{\alpha_0} a_1^{l_1} a_2^{l_2} A_1^{\alpha_1} A_2^{\alpha_2},\ v(w) = a_3^{l_3} a_4^{l_4} A_3^{\alpha_3} A_4^{\alpha_4}$$

where $\alpha_j \in \{0, 1\}$, $l_j \geq 0$. Note that if two non-zero words w, w' are equal in $\tilde{S}(MM_4)$, then $u(w)$ and $u(w')$ are equal in $\tilde{S}(MM_2)$.

We claim that $\tilde{S}(MM_4)$ is residually finite. Indeed, consider two words w_1, w_2 in the generators of $\tilde{S}(MM_4)$ which are not equal in $\tilde{S}(MM_4)$.

Suppose first that w_1 does not contain a q-letter. Then consider the ideal I of $\tilde{S}(MM_4)$ generated by all q-letters. The Rees factor-semigroup $S' = \tilde{S}(MM_4)/I$ is generated by letters $a_1, \ldots, a_4, A_1, \ldots, A_4$ subject to commutativity relations and 0-relations. Therefore every non-zero element of S' has finitely many divisors, hence S' is residually finite by Lemma 3.11. Since the inequality $w_1 \neq w_2$ survives passing to the Rees quotient, we can separate w_1 and w_2 by a homomorphism onto a finite semigroup.

Thus we can assume that both w_1 and w_2 start with a q-letters. For every

$$w = q_i a_1^{l_1} a_2^{l_2} A_1^{\alpha_1} A_2^{\alpha_2} a_3^{l_3} a_4^{l_4} A_3^{\alpha_3} A_4^{\alpha_4}$$

where $\alpha_i = \alpha_i(w_1) \in \{0, 1\}$ we denote l_i by $l_i(w)$, and α_i by $\alpha_i(w)$.

Then for every word w' obtained from w by applying the relations of $\tilde{S}(MM_4)$ we have $\alpha_j(w') = \alpha_j(w)$, $j = 1, 2, 3, 4$, $u(w)$ is obtained from $u(w_1)$ by applying relations of $S(MM_2)$.

Claim There are only finitely many words that are equal to $u(w_1)$ in $\tilde{S}(MM_2)$.

Indeed, if one of the numbers α_1 or α_2 is 0, the Claim is true by Theorem 2.7 (c). If both α_1 and α_2 are equal to 1, since w_1 is not equal to 0 in $\tilde{S}(MM_4)$, the configuration $(i; l_1, l_2)$ is not accepted by MM_2 (here we use the relation $q_0 = 0$), and the Claim is true because MM_2 is sym-universally halting.

Let Y be the set of all words that are equal to w_1 in $\tilde{S}(MM_4)$. Then the Claim implies that the set of numbers $l_3(w) - l_4(w)$, $w \in Y$, is finite. Let $D(w_1)$ be the maximum of all these numbers. The number $D(w_2)$ is defined similarly. Let D be the maximum of $D(w_1)$, $D(w_2)$.

Let us add the relations $a_3^D = a_3^{2D}, a_4^D = a_4^{2D}$ to $\tilde{S}(MM_4)$. Let $\bar{S}$ be the resulting semigroup, and $\psi : \tilde{S}(MM_4) \to \bar{S}$ be the corresponding homomorphism. Then it is easy to see that $\psi(w_1) \neq \psi(w_2)$. Notice that in $\bar{S}$, every non-zero element has finite number of divisors. Indeed, it is true for $\tilde{S}(MM_2)$ (see Remark 3.10) and the number of different elements of $\bar{S}$ of the form $v(w)$ is finite. Hence we can again use Lemma 3.11.

The function $\rho(n)$ for the semigroup $\tilde{S}(MM_4)$ is at least as large as the following function $\Psi(n)$ associated with the machine MM_4: $\Psi(n)$ is the smallest number such that for every non-accepted input configuration of M of length $\leq n$, the machine MM_4 halts after at most $\Psi(n)$ steps (we call this function the *co-time* function of MM_4). Indeed let c be an input configuration of length at most n such that MM_4 halts after exactly $\Psi(n)$ steps starting at c. Suppose that the word $w(c)$ in $\tilde{S}(MM_4)$ corresponding to the configuration c can be separated from 0 in a homomorphic image E of $\tilde{S}(MM_4)$ with at most $\Psi(n) - 1$ elements. Then the images of a_3, a_4 in that semigroup satisfy $z^D = z^{2D}$ for some $D < T(n)$. Note that

- the halting computation has $> D$ steps,
- the letter a_3 does not occur in $w(c)$,
- every old command of MM_4 adds one coin in glass 3,

Therefore there exists a word W which is equal to $w(c)$ in $\tilde{S}(MM_4)$ and which has the form

$$q_j a_1^{l_1} a_2^{l_2} A_1 A_2 a_3^D A_3 A_4.$$

Modulo relations corresponding to the commands (4), this word is equal to

$$q_j a_1^{l_1} a_2^{l_2} A_1 A_2 a_3^{2D} A_3 a_4^D A_4.$$

The image of the latter word in E is equal to

$$q_j a_1^{l_1} a_2^{l_2} A_1 A_2 a_3^D A_3 a_4^D A_4$$

which, again modulo the relations corresponding to the commands (4), is equal to

$$q_j a_1^{l_1} a_2^{l_2} A_1 A_2 A_3 A_4.$$

Here $j > 1$ by our assumption that command number 1 cannot be used in the middle of a computation consisting of old commands. Hence the latter word is equal to 0 by the relations corresponding to the commands (5) and the relation $q_0 = 0$, a contradiction.

Note that the co-time function of a Turing machine recognizing a recursive set can be larger than any given recursive function. Indeed, we can assume that the Turing machine has a history tape as in Sect. 2.2, so if the machine does not accept the input, the last configuration in the halting computation has the history of computation written on the tape. Then after the Turing machine halts without accepting, we can make it compute some large recursive function, taking as a variable the word written on the history tape. It remains to note that the co-time function of a Minsky machine simulating that Turing machine cannot be smaller. □

In the next section, we shall use the following properties of the semigroups studied in this section. Let $\check{S}$ be the semigroup given by all non-Minsky defining relations of $S(MM_k)$ and let $\check{\bar{S}}$ be the semigroup given by all non-Minsky defining relations of the semigroup $\bar{S}$ from the proof of Theorem 3.12.

Lemma 3.15 (a) *The growth function of $\check{S}$ is polynomial of degree k.*

(b) *The semigroup $\check{S}$ satisfies the following property:*

(P) If

$$w = q_i^{\alpha} a_1^{m_1} \dots a_k^{m_k} A_1^{\beta_1} \dots A_k^{\beta_k},$$

and

$$w' = q_{i'}^{\alpha'} a_1^{m'_1} \dots {}_k^{m'_k} A_1^{\beta'_1} \dots A_k^{\beta'_k},$$

where $\alpha, \alpha', \beta_j, \beta'_j \in \{0, 1\}$ and $w = w'$ in $\check{S}$, then $q_i^{\alpha} = q_{i'}^{\alpha'}$ (i.e., either $\alpha = \alpha' = 0$ or $i = i'$ and $\alpha = \alpha'$), $m_j = m'_j$, $\beta_j = \beta'_j$, $j = 1, \dots, k$.

(c) *The semigroup $\check{\bar{S}}$ satisfies the following two properties*

(Q1) If

$$w = q_i^{\alpha} a_1^{m_1} \dots a_k^{m_k} A_1^{\beta_1} \dots A_k^{\beta_k},$$

and

$$w' = q_{i'}^{\alpha'} a_1^{m'_1} \dots a_k^{m'_k} A_1^{\beta'_1} \dots A_k^{\beta'_k},$$

where $i, i' \neq 0$, $\alpha, \alpha', \beta_j, \beta'_j \in \{0, 1\}$, *and* $w = w'$ *in* $\overset{\vee}{\overset{\vee}{S}}$, *then* $q_i^{\alpha} = q_{i'}^{\alpha'}$, $\beta_j = \beta'_j$, $j = 1, \ldots, k$.

(Q2) For every word w *the equality* $wA_i = 0$ *in* $\overset{\vee}{\overset{\vee}{S}}$ *implies* $wa_iA_i = 0$ *in* $\overset{\vee}{\overset{\vee}{S}}$.

Proof (a) Indeed every non-zero element of length $\leq n$ of $\overset{\vee}{S}$ is represented (modulo the commutativity relations) by a word of the form $q_i a_1^{m_1} \ldots a_k^{m_k} A_1^{\epsilon_1} \ldots A_k^{\epsilon_k}$ where $m_j \leq n, \epsilon_j \in \{0, 1\}$.

To prove Properties (P) and (Q1) notice that the exponents of q_i, a_j, A_j do not change when we apply the defining relations of these semigroups to w.

To prove (Q2) notice that word the $w = q_i^{\alpha} a_1^{m_1} \ldots a_k^{m_k} A_1^{\beta_1} \ldots A_k^{\beta_k}$ is equal to 0 in $\overset{\vee}{\overset{\vee}{S}}$ if and only if $i = 0$ and then apply (Q1). □

4 Simulation of Minsky machines in solvable groups

Recall that a *variety* of algebraic structures is a class of all algebraic structures of a given signature satisfying a given set of *identities* (also called *laws*). Equivalently, by a theorem of Birkhoff [35] a variety is a class of algebraic structures closed under taking cartesian products, homomorphic images and substructures. Every variety contains free objects (called *relatively free* algebraic structures). One can define algebraic structures that are finitely presented in a variety as factor-structures by congruence relations generated by finite number of equalities. Every finitely presented algebraic structure which belongs to a variety $\mathcal{V}$ is finitely presented inside $\mathcal{V}$ but the converse is very rarely true. See [32] for a survey of algorithmic problems for varieties of different algebraic structures (mostly semigroups, groups, associative and Lie algebras). In this section we concentrate on varieties of groups (see [43]). The most well known varieties are the variety of Abelian groups $\mathcal{A}$ given by the identity $[x, y] = 1$, the variety of nilpotent groups of class c, $\mathcal{N}_c$ given by the identity $[\ldots[x_1, x_2], \ldots, x_{c+1}] = 1$, etc. The class of Abelian groups of finite exponent d, $\mathcal{A}_d$, is also a variety, given by two identities $[x, y] = 1, x^d = 1$.

If $\mathcal{U}$ and $\mathcal{V}$ are two varieties of groups then the class of groups consisting of extensions of groups from $\mathcal{U}$ by groups from $\mathcal{V}$ is again a variety (the *product* of $\mathcal{U}$ and $\mathcal{V}$) denoted by $\mathcal{UV}$. The product of varieties is associative [43]. For example the variety of all solvable groups of class c is the product of c copies of the variety $\mathcal{A}$. If $\mathcal{V}$ is a variety of groups, then $\mathcal{ZV}$ is the variety consisting of all central extensions of groups from $\mathcal{V}$. For example $\mathcal{N}_2 = \mathcal{ZA}$ and, more generally, $\mathcal{N}_{c+1} = \mathcal{ZN}_c$ for every $c \geq 1$.

4.1 The construction

Let MM_k be a Minsky machine with k glasses and $N + 1$ commands (numbered $0, \ldots, N$). We are going to construct a group $G(MM_k)$ simulating MM_k. The group $G(MM_k)$ will be very close to the semigroup $S(MM_k)$ constructed above. The main idea will be to replace the product in $S(MM_k)$ by another, derived, operation and make sure that with respect to the new operation the semigroup $S(MM_k)$ "embeds" into our group, in such a way that the "image" of $S(MM_k)$ is "almost the whole" group.

The group will be generated by the x-letters which will be related to the letters q_i from $S(MM_k)$, and also a-letters $a_1, \dots, a_k$, A-letters $A_1, \dots, A_k$ and some other a- and A-letters that help us impose the necessary commutativity relations that, in particular, make the group solvable, and contain "very few" extra elements. The group we are going to construct will be a semidirect product of an elementary Abelian normal subgroup generated (as a normal subgroup) by the x-letters by a semidirect product of an Abelian subgroup generated (as a normal subgroup) by A-letters and an Abelian subgroup generated by a-letters.

Thus we should have a way to ensure that in a subgroup generated by two sets of letters $Z \cup Y$, the normal subgroup generated by Z is Abelian. This is done with the help of the following lemma due to Baumslag [3] and Remeslennikov [51]. In that lemma we denote $u^a = a^{-1}ua$ and $u^{a+b} = u^a u^b$ (note that although u^{a+b} is not necessarily equal to u^{b+a}, the equality will hold if the normal subgroup generated by u is Abelian, which is going to be the case every time we apply this lemma).

Lemma 4.1 ([3,51]). *Suppose that a group H is generated by three sets X, $F = \{a_i \mid i = 1, \dots, m\}$, $F' = \{a_i' \mid i = 1, \dots, m\}$ such that*

(1) *The subgroup generated by $F \cup F'$ is Abelian;*
(2) *For every $a \in F$ and every $x \in X$ we have $x^{f(a)} = x^{a'}$ for some monic polynomial f of a which has at least two terms (everywhere below $f(t) = t - 1$ and hence $x^{f(a)} = a^{-1}xax^{-1} = [a, x^{-1}]$, so we will not mention f again);*
(3) *$[x_1^{a_1^{\alpha_1} \dots a_m^{\alpha_m}}, x_2] = 1$, for every $x_1, x_2 \in X$, and every $\alpha_1, \dots, \alpha_m \in \{0, 1, -1\}$. Then the normal subgroup generated by X in the group $H = \langle X \cup F \cup F' \rangle$ is Abelian, and H is metabelian.*

If the elements a_i and a_i' and the set X satisfy the conditions of Lemma 4.1 we will call a_i', BR-conjoints to a_i with respect to X (and the polynomial f), $i = 1, \dots, m$.

Consider the free commutative monoid generated by letters $A_0, \dots, A_k$. Let U' be the set of all divisors of the element $A_0 A_1 \dots A_k$ in that monoid, and U be the set of all symbols $q_j u$, $u \in U'$, $j = 0, \dots, N$. Also fix a prime p (say, $p = 2$).

The generating set of our group $G = G(MM_k)$ will consist of three subsets:

$$
\begin{aligned}
L_0 &= \{x(u) \mid u \in U, \quad i = 0, \dots, N\};\\
L_1 &= \{A_i \mid \quad i = 0, \dots, k, \};\\
L_2 &= \{a_i, a_i', \tilde{a}_i, \tilde{a}_i' \mid \quad i = 1, \dots, k\}.
\end{aligned}
$$

We introduce notation for some subgroups of the group G. Denote $H_i = \langle L_i \rangle$, $i = 0, 1, 2$. Denote also

$$
M_0 = \{\tilde{a}_i, \tilde{a}_i', A_0 \mid i = 1, \dots, k\}, \ M_i = \{a_i, a_i', A_i\}, i = 1, \dots, k.
$$

The group $G(MM_k)$ has the following finite set of defining relations:

(G1) Relations saying that H_0 and H_1 are Abelian groups of exponent p, and H_2 is an Abelian group.
(G2) Any $y \in M_i, z \in M_j, i \neq j \in \{0, \dots, k\}$, commute.

(G3) For every $i = 1, \ldots, k$, $(a_i')^{-1}$ is a BR-conjoint to a_i^{-1} with respect to $\{A_i\}$.
(G4) The elements of the set $\{(\tilde{a}_i')^{-1} \mid i = 1, \ldots, k\}$ are BR-conjoints to elements of the set $\{\tilde{a}_i^{-1} \mid i = 1, \ldots, k\}$ with respect to $\{A_0\}$.
(G5) a) If $u \in U$ does not contain A_i for some $i = 0, \ldots, k$, then $[x(u), A_i] = x(uA_i)$.
b) For every $i = 1, \ldots, k$, if u does not contain A_i, then $x(u)^{a_i - 1} = x(u)^{a_i'}$ (where $x^{a-1} = x^a x^{-1}$),
c) For every $i = 0, \ldots, k$, if u contains A_i, $z \in M_i$, then $[x(u), z] = 1$.
(G6) $x(q_j)^{a_i} = x(q_j)^{\tilde{a}_i}$, $x(q_j)^{a_i'} = x(q_j)^{\tilde{a}_i'}$, $j = 0, \ldots, N$, $i = 1, \ldots, k$.
(G7) $[x(u)^z, x(v)] = 1$, where $z = a_1^{\alpha_1} \ldots a_k^{\alpha_k}$, $\alpha_i \in \{-1, 0, 1\}$, $u, v \in U$

Remark 4.2 Relations (G7) together with (G1) and (G5) b) imply that for every subset $I \subseteq \{1, \ldots, k\}$ the letters $\{a_i' \mid i \in I\}$ are BR-conjoints of $\{a_i \mid i \in I\}$ with respect to the set of all $x(u)$'s where u does not contain letters A_i, $i \in I$.

(G8) Relations constructed from the program of the machine MM_k. For every $f \in G$ denote

$$f * a_i = f^{-1} f^{a_i} f^{-a_i^{-1}} f^{(a_i')^{-1}}, \quad i = 1, \ldots, k,$$

also let

$$f * A_i = [f, A_i], \quad i = 0, \ldots, k.$$

We denote $(\ldots (t_1 * t_2) * \ldots) * t_m$ by $t_1 * \ldots * t_m$, and $t_1 * \underbrace{t_2 * \ldots * t_2}_{n}$ by $t_1 * t_2^{(n)}$.

The relations corresponding to the commands of MM_k are in the following table.

Command of M	Relation of $G(MM_k)$
$i \to \text{Add}(n_1, \ldots, n_m); j$	$x(q_i A_0) = x(q_j A_0) * a_{n_1} * \ldots * a_{n_m}$
$i, \epsilon_{n_1} > 0, \ldots, \epsilon_{n_m} > 0 \to \text{Sub}(n_1, \ldots, n_m); j$	$x(q_i A_0) * a_{n_1} * \ldots * a_{n_m} = x(q_j A_0)$
$i, \epsilon_{n_1} = 0, \ldots, \epsilon_{n_m} = 0 \to j$	$x(q_i A_0) * A_{n_1} * \ldots * A_{n_m} = x(q_j A_0) * A_{n_1} * \ldots * A_{n_m}$

(6)

Theorem 4.3 (a) *The group $G(MM_k)$ belongs to $\mathcal{A}_p^2\mathcal{A} \cap \mathcal{ZN}_{k+1}\mathcal{A}$.*
(b) *The equality*

$$x(q_i A_0) * a_1^{(m_1)} * \ldots a_k^{(m_k)} * A_1^{(\alpha_1)} * \ldots * A_k^{(\alpha_k)} = x(q_j A_0) * a_1^{(n_1)} * \ldots * a_k^{(n_k)} * A_1^{(\beta_1)} * \ldots A_k^{(\beta_k)}$$

where $\alpha_i, \beta_i \in \{0, 1\}$ is true in $G(MM_k)$ if and only if the equality

$$q_i a_1^{m_1} \ldots a_k^{m_k} A_1^{\alpha_1} \ldots A_k^{\alpha_k} = q_j a_1^{n_1} \ldots a_k^{m_k} A_1^{\beta_1} \ldots A_k^{\beta_k}$$

is true in the semigroup $S(MM_k)$ (in particular, $\alpha_i = \beta_i$ for every i).

(c) *The equality*

$$x(q_i) * a_1^{(m_1)} * \dots a_k^{(m_k)} * A_1^{(\alpha_1)} * \dots * A_k^{(\alpha_k)} = x(q_j) * a_1^{(n_1)} * \dots * a_k^{(n_k)} * A_1^{(\beta_1)} * \dots A_k^{(\beta_k)}$$

where $\alpha_i, \beta_i \in \{0, 1\}$ *is true in* $G(MM_k)$ *if and only if the equality*

$$q_i a_1^{m_1} \dots a_k^{m_k} A_1^{\alpha_1} \dots A_k^{\alpha_k} = q_j a_1^{n_1} \dots a_k^{m_k} A_1^{\beta_1} \dots A_k^{\beta_k}$$

is true in the semigroup $\hat{S}$ (*i.e., these words coincide, see Lemma* 3.15 (*b*)).

Proof The proof of Part (a) is divided into several lemmas.

Lemma 4.4 *The subgroup* $\langle H_1 \cup H_2 \rangle$ *of* G *is metabelian and a semidirect product of the Abelian normal subgroup* $H_1^{H_2}$ *of exponent* p, *and* H_2.

Proof Indeed by relations (G2),

$$\langle M_i, i = 0, \dots, k \rangle = \prod_{i=1}^{k} \langle a_i, a_i', A_i \rangle \times \langle \tilde{a}_i, \tilde{a}_i', A_0, i = 1, \dots, k \rangle.$$

Using relations (G1), (G3), (G4), we can apply Lemma 4.1 to each of the factors in that direct product and conclude that each of them is metabelian and a semidirect product of the Abelian normal subgroup of exponent p generated by the intersection of $\{A_i \mid i = 0, \dots, k\}$ with that factor, and the Abelian group generated by the a-letters from that factor. □

Lemma 4.5 *The normal subgroup* T *of* G *generated as a normal subgroup by all the elements* $x(u), u \in U$, *is Abelian of exponent* p.

Proof Relations (G5) a) of the group G imply that every element $x(u), u \in U$, is a product of elements $x(q_j)^z, z \in H_1, i = 0, \dots, N$. Therefore, it is enough to show that

$$x(q_k)x(q_t)^z = x(q_t)^z x(q_k) \tag{7}$$

for any $z \in \langle H_1, H_2 \rangle$ and any k, t. To simplify these equalities notice that $z = z_0 z_1 \dots z_k$ where $z_i \in \langle M_i \rangle$ by (G2). Therefore equalities (7) are equivalent to

$$x(q_k)^{z_0} x(q_t)^{z_1 \dots z_k} = x(q_t)^{z_1 \dots z_k} x(q_k)^{z_0}. \tag{8}$$

We can represent element $x(q_j)^{z_i}$, $i \geq 1$, as a product of elements of the form $x(q_j)^{a_i^p (a_i')^q}$ and $x(q_j A_i)^{\tilde{a}_i^p (\tilde{a}_i')^q}$. Indeed we have the following sequence of equalities deduced using (G2), (G5), (G6):

$$x(q_j)^{a_i^{r_1}(a_i')^{s_1}A_i^{t_1}\ldots a_i^{r_s}(a_i')^{s_k}A_i^{t_k}} \overset{\text{(G6)}}{=\!=} x(q_j)^{\tilde{a}^{r_1}(\tilde{a}_i')^{s_1}A^{t_1}\ldots a_i^{r_k}(a_i')^{s_k}A_i^{t_k}}$$

$$\overset{\text{(G2)}}{=\!=} x(q_j)^{A_i^{s_1}\tilde{a}_1^{r_1}(\tilde{a}_i')^{s_1}a_i^{r_2}(a_i')^{s_2}\ldots a_i^{r_k}(a_i')^{s_k}A_i^{t_k}}$$

$$\overset{\text{(G5) a), c), (G6)}}{=\!=\!=\!=\!=\!=\!=} x(q_j)^{a_i^{r_1+r_2}(a_i')^{s_1+s_2}A_i^{t_2}\ldots a_i^{r_k}(a_i')^{s_k}A_i^{t_k}(x(q_jA_i)^{t_1})^{\tilde{a}_i^{r_1}(\tilde{a}_i')^{s_1}}} =$$

$$\ldots = x(q_j)^{a_i^{r_1+\ldots+r_k}(a_i')^{s_1+\ldots+s_k}(x(q_tA_i)^{t_k})^{\tilde{a}_i^{r_1+\ldots r_k}(\tilde{a}_i')^{s_1+\ldots s_k}}\ldots(x(q_tA_i)^{t_1})^{\tilde{a}_i^{r_1}(\tilde{a}_i')^{s_1}}}. \tag{9}$$

Repeating this argument k times, one proves that $x(q_j)^{z_1z_2\cdots z_k}$ can be represented as a product of elements of the form $x(u)^y$ where $u \in U$, $y \in H_2$. A similar proof (using also (G4)) gives that $x(q_j)^{z_0}$ is a product of elements of that form. It remains to note that elements of the form $x(u)^y$, $u \in U$, $y \in H_2$ commute by Remark 4.2 and Lemma 4.1. □

Remark 4.6 Note that equalities (9) and similar equalities when $x(q_j)$ is replaced by $x(u)$, $u \in U$, imply the following: if y is a product of elements of the form $a_i^{r_l}(a_i')^{s_l}A_i$ and $\sum_l r_l = \sum_l s_l = 0$, then $[x(u), y]$ is equal to 1 if u contains A_i or is equal to a product of conjugates of elements $x(uA_i)$ by elements from $\langle\tilde{a}_i\rangle \times \langle\tilde{a}_i'\rangle$ otherwise. Similarly, suppose that y is a product of elements from M_0, each factor containing A_0, and the total exponent of every $\tilde{a}_i$ (resp. $\tilde{a}_i'$) is 0. Then $[x(u), y] = 1$ provided u contains A_0 and is a product of conjugates of $x(uA_0)$ by elements from $\langle a_i, a_i'\rangle$ provided u does not contain A_0.

By construction, the group G is a semidirect product of T and the metabelian group $H_1^{H_2} \rtimes H_2$. By Lemma 4.4, G is solvable of class 3 and, moreover, belongs to $\mathcal{A}_p^2\mathcal{A}$.

Remark 4.7 The proof of Lemma 4.5 shows that T is generated (as an Abelian group) by elements of the form $x(u)^y$ where $u \in U$ and $y \in H_2$.

Lemma 4.8 *The quotient of $G(MM_k)$ over the center satisfies the identity*

$$[[x_1, y_1], [x_2, y_2], \ldots, [x_{k+2}, y_{k+2}]] = 1.$$

This means that G belongs to the variety $\mathcal{ZN}_{k+1}\mathcal{A}$.

Proof Let P be the derived subgroup of $G(MM_k)$. By Lemma 4.5, every element of P is a product of an element of T and an element of $H_1^{H_2}$. It also follows from Lemma 4.5 that $[P, P] \subseteq T$, hence by Remark 4.7, it is generated by elements of the form $x(u)^y$, $u \in U$, $y \in H_2$, the word u contains at least one A_i, $i = 0, \ldots, k$. Since T is Abelian, the subgroup $\underbrace{[P, P, \ldots, P]}_{k+2}$ is generated by the commutators

$$\left[x(u)^y, h_{1,1}^{h_{2,1}}, \ldots, h_{1,k}^{h_{2,k}}\right]$$

for some $h_{1,i} \in H_1$, $y, h_{2,i} \in H_2$. An easy induction shows that every such commutator is a conjugate of

$$\left[x_u, h_{1,1}^{y'}, \ldots, h_{1,k}^{y'}\right] \tag{10}$$

where $y' \in H_2$.

Let $h \in H_1, u \in U, y \in H_2$. Suppose that $h = A_{i_1}^{t_1} \cdot \ldots \cdot A_{i_s}^{t_s}$ where $t_i \neq 0$. Consider $[x(u), h^y]$. Then Remark 4.6 implies that $[x(u), h^y]$ is a product of elements of the form $x(u')^{y'}$ where $u' \in U$ contains letters $A_{i_1}, \ldots, A_{i_s}$ and it may not be equal to 1 only if one of the letters A_{i_j} does not occur in u. Therefore the commutator (10) is either equal to 1 or is a product of elements of the form $x(u')^{y''}$ where the word $u' \in U$ contains all letters $A_0, A_1, \ldots, A_k$, $y'' \in H_2$. But every such $x(u')$ is in the center of $G(MM_k)$ by (G5) c). Hence $\underbrace{[P, \ldots, P]}_{k+2}$ is contained in the center of $G(MM_k)$. □

We now prove Parts (b) and (c) of Theorem 4.3. For every configuration $c = (i; m_1, \ldots, m_k)$ of MM_k let $w_G(c) = x(q_i A_0) * a_1^{(m_1)} * \ldots a_k^{(m_k)} * A_1^{(\alpha_1)} * \ldots * A_k^{(\alpha_k)}$. To prove (b), as we mentioned before Property 3.1, we need to prove Properties 3.1 and 3.2. Property 3.1 for $G(MM_k)$ is proved in the same way as for the semigroup $S(MM_k)$ (see [32,55]),since the only property of $S(MM_k)$ used there was that the word $w = q_i a_1^{l_1} \ldots a_k^{l_k} A_1^{\alpha_1} \ldots A_k^{\alpha_k}$ is equal in $S(MM_i)$ to any word obtained from w by permuting a_i with a_j, A_i with A_j and a_i with A_j $(i \neq j)$. The same is true for words of the form $w_G(c)$ in $G(MM_k)$ by the definition of the operation $*$, relations (G1), (G2) and Lemma 4.5.

In order to prove Property 3.2 we will define a new group $\overset{\diamond}{G}$ that is an image of G under some homomorphism that is injective on the elements from Parts (b) and (c).

Let $\check{S}$ be the semigroup with the same generating set as $S(MM_k)$ subject all the relations of $S(MM_k)$ except the Minsky relations (3). That semigroup does not depend on MM_k. Thus non-zero elements in $\check{S}$ have the form

$$q_i^{\alpha_1} a_1^{l_1} \ldots a_k^{l_k} A_1^{\alpha_1} \ldots A_k^{\alpha_k}$$

where $l_j \in \mathbb{N}, \alpha_j \in \{0, 1\}$. Let W be the set of all non-zero elements of $\check{S}$ containing a q-letter.

Let ψ be the natural homomorphism from $\check{S}$ onto $S(MM_k)$. Let $W_0 = \psi(W)$. We will need the set of vectors $\{1, 2, 3\}^k \subset \mathbb{Z}^k$ with coordinates $1, 2, 3$. Its elements will be denoted by $\vec{i}$. The j-th coordinate of the vector $\vec{i}$ will be denoted by $\vec{i}_j$, the standard unit vectors $(0, \ldots, 1, \ldots, 0)$ are denoted by e_j. Consider the set of symbols $\{z(\vec{i}, u) \mid \vec{i} \in \{1, 2, 3\}^k, u \in W \cup W_0\}$ and the multiplicative Abelian group T_1 of exponent p freely generated by this set.

For each letter from $L_1 \cup L_2$, we define an automorphism of T_1. The group $\overset{\diamond}{G}$ will be the semidirect product of T_1 and the group generated by these automorphisms. For simplicity we will denote automorphisms corresponding to letters from $L_1 \cup L_2$ by the same letters (the automorphisms are just conjugations by these letters).

Let us start with automorphisms a_j, a_j'. We have to define $z(\vec{i}, w)^{a_j}$ and $z(\vec{i}, w)^{a_j'}$ for every $\vec{i}$ and every $w \in W \cup W_0$. This definition does not depend on whether w belongs to W or W_0. First suppose that w does not contain A_j. Then

$$z(\vec{i},w)^{a_j}=\begin{cases} z(\vec{i},w)z(\vec{i}+\vec{e}_j,w)z(\vec{i}+2\vec{e}_j,w)z(\vec{i},wa_j) & \text{if} \quad \vec{i}_j=1;\\ z(\vec{i},w)z(\vec{i}-\vec{e}_j,w)^{-1} & \text{if} \quad \vec{i}_j=2;\\ z(\vec{i}-2\vec{e}_j,w) & \text{if} \quad \vec{i}_j=3. \end{cases}$$

$$z(\vec{i},w)^{a_j'}=z(\vec{i},w)^{-1}z(\vec{i},w)^{a_j}. \tag{11}$$

If w contains letter A_j, then let $z(\vec{i},w)^{a_j}=z(\vec{i},w)^{a_j'}=z(\vec{i},w)$.

It is easy to prove that a_j is an automorphism by constructing the automorphism a_j^{-1} (view (11) as a triangular system of linear equations and solve it by backward substitution). For example:

$$z(\vec{i},w)^{a_j^{-1}}=\begin{cases} z(\vec{i}-\vec{e}_j,w)^{-1}z(\vec{i},w)^{-1}z(\vec{i},wa_j)^{-1}, & \text{if} \quad \vec{i}_j=3\\ z(\vec{i},w)z(\vec{i}+\vec{e}_j,w), & \text{if} \quad \vec{i}_j=2\\ z(\vec{i}+2\vec{e}_j,w), & \text{if} \quad \vec{i}_j=1. \end{cases}$$

The automorphism $\tilde{a}_j$ is defined similarly. But the definition depends on whether w is in W_0 or W. If $w\in W_0$ then $z(\vec{i},w)^{\tilde{a}_j}=z(\vec{i},w)$. If $w\in W$ and does not contain A_j then $z(\vec{i},w)^{\tilde{a}_j}=z(\vec{i},w)^{a_j}$.

If $w\in W$ and w contains A_j, i.e., $w=vA_j$ in $\check{S}$ for some v, then

$$z(\vec{i},w)^{\tilde{a}_j}=\begin{cases} z(\vec{i},w)z(\vec{i}+\vec{e}_j,w)z(\vec{i}+2\vec{e}_j,w)z(\vec{i},va_jA_j), & \text{if} \quad \vec{i}_j=1;\\ z(\vec{i},w)z(\vec{i}-\vec{e}_j,w)^{-1}, & \text{if} \quad \vec{i}_j=2;\\ z(\vec{i}-2\vec{e}_j,w), & \text{if} \quad \vec{i}_j=3. \end{cases}$$

$$z(\vec{i},w)^{\tilde{a}_j'}=z(\vec{i},w)^{-1}z(\vec{i},w)^{\tilde{a}_j}.$$

The automorphisms corresponding to A_j, $j=1,\ldots,k$, are defined as follows:

$$z\left(\vec{i},w\right)^{A_j}=z\left(\vec{i},w\right)z\left(\vec{i},wA_j\right)$$

if $w\in W\cup W_0$ does not contain A_j and

$$z\left(\vec{i},w\right)^{A_j}=z\left(\vec{i},w\right)$$

if w contains A_j.

Finally the automorphism corresponding to A_0 is defined as follows:

$$z\left(\vec{i},w\right)^{A_0}=z\left(\vec{i},w\right)z\left(\vec{i},\psi(w)\right)$$

if $w \in W$ and

$$z\left(\vec{i}, w\right)^{A_0} = z\left(\vec{i}, w\right)$$

if $w \in W_0$.

The following lemma is obtained by a straightforward application of the definition of the automorphisms above and the definition of the operation $*$. This lemma implies that $\overset{\diamond}{G}$ satisfies (G8) if we replace $x(u)$ by $z(\vec{1}, u)$ where $\vec{1}$ is the vector $(1, 1, \ldots, 1)$ (since the corresponding relations hold in $S(MM_k)$).

If $w \in W \cup W_0$, $d \in \{a_j, A_j \mid j = 1, \ldots, k\}$ then by wd we mean the product of w and d in $S(MM_k)$ provided $w \in W_0$, or in $\hat{S}$ provided $w \in W$.

The proof of the following lemma is by inspection of various cases and mostly left to the reader.

Lemma 4.9 *The following relations hold in $\overset{\diamond}{G}$. For every $d \in \{a_j, A_j \mid j = 1, \ldots, k\}$, $w \in W \cup W_0$*

$$z(\vec{1}, w) * d = z(\vec{1}, wd)$$

where $*$ is defined in (G8). Here we set $z(\vec{1}, 0) = 1$ (where 0 is the zero element in $S(MM_k)$ or $\hat{S}$, 1 in the right hand side is the identity element in $G(MM_k)$).

Proof For example, if $d = A_1$ then $z(\vec{1}, w) * d = [z(\vec{1}, w), A_1]$. It is equal to 1 if w contains A_1, and it is equal to $z(\vec{1}, w)^{-1} z(\vec{1}, w)^{A_1} = z(\vec{1}, w)^{-1}(z(\vec{1}, w) z(\vec{1}, wA_j)) = z(\vec{1}, wA_j)$ if w does not contain A_1. Thus in both cases $z(\vec{1}, w) * d = z(\vec{1}, wd)$. □

By definition, $\overset{\diamond}{G}$ is the semidirect product of T_1 and the subgroup of $\mathrm{Aut}(T_1)$ generated by the automorphisms corresponding to the elements from $L_1 \cup L_2$. From the definition of the automorphisms and Lemma 4.9, it follows that $\overset{\diamond}{G}$ is generated by the elements $z(\vec{1}, u)$, $u \in U$, and the automorphisms corresponding to elements of $L_1 \cup L_2$. It is easy to check that all the relations (G1)-(G8) hold in $\overset{\diamond}{G}$, therefore

Lemma 4.10 *The map that sends every a- or A-letter to itself, every $x(u)$ with $u \in U$ containing A_0, $u = vA_0$, to $z(\vec{1}, \psi(v))$ and every $x(u)$ with u not containing A_0 to $z(\vec{1}, u)$ extends to a homomorphism γ from G to $\overset{\diamond}{G}$.*

Lemma 4.11 *The homomorphism γ is surjective and the preimage of T_1 is T.*

Proof We only need to define pre-images of elements $z(\vec{i}, w) \in \bar{G}$, $w \in W \cup W_0$. By the definition of γ, we have $\gamma(x(u)) = z(\vec{1}, u)$ for every $u \in U$ which does not contain A_0, and $\gamma(x(u'A_0)) = z(\vec{1}, \psi(u'))$ so $x(u)$, $u \in U$, are preimages of all $z(\vec{1}, u)$. The preimage $x(\vec{i}, w')$ of $z(\vec{i}, w)$ for every $\vec{i}$ and w is defined by induction on the length of w and the sum of $\vec{i}_j$ (the base of induction, where $\vec{i} = \vec{1}$ is obvious).

Suppose that w does not contain A_j, $j = 1, \ldots, k$, and $\vec{i}_j = 1$, $\vec{i}'$ is arbitrary. Then we define:

$$x(\vec{i} + \vec{e}_j, w) = x(\vec{i}, w)^{-(a'_j)^{-1}},$$
$$x(\vec{i} + 2\vec{e}_j, w) = x(\vec{i}, w)^{a_j^{-1}},$$
$$x(\vec{i}', wa_j) = x(\vec{i}', w) * a_j.$$

We also set $x(\vec{i}, wA_j) = x(\vec{i}, w) * A_j$ for any $\vec{i}$. This covers the case when w contains A_j and $\vec{i}_j = 2$ or 3.

It is easy to see that for every $\vec{i}$ and $w \in W \cup W_0$, we have $\gamma(x(\vec{i}, w)) = z(\vec{i}, w)$. This proves the lemma. □

In $\overset{\diamond}{G}(MM_k)$, consider the set P_0 of elements

$$z(\vec{1}, \psi(q_i)) * a_1^{(m_1)} * \ldots a_k^{(m_k)} * A_1^{(\alpha_1)} * \ldots * A_k^{(\alpha_k)} \tag{12}$$

where $\alpha_i \in \{0, 1\}$ and the set P of elements

$$z(\vec{1}, q_i) * a_1^{(m_1)} * \ldots a_k^{(m_k)} * A_1^{(\alpha_1)} * \ldots * A_k^{(\alpha_k)} \tag{13}$$

By construction $P \cap P_0 = \emptyset$, elements (12) are different if and only if elements

$$q_i a_1^{m_1} \ldots a_k^{m_k} A_1^{\alpha_1} \ldots A_k^{\alpha_k}$$

from $S(MM_k)$ are different, and elements (13) are different if and only if the corresponding elements $q_i a_1^{m_1} \ldots a_k^{m_k} A_1^{\alpha_1} \ldots A_k^{\alpha_k}$ of $\check{S}$ are different. By Lemma 4.9 the element $x(q_i A_0) * a_1^{(m_1)} * \ldots a_k^{(m_k)} * A_1^{(\alpha_1)} * \ldots * A_k^{(\alpha_k)}$ is equal to the element (12) and the element $x(q_i) * a_1^{(m_1)} * \ldots a_k^{(m_k)} * A_1^{(\alpha_1)} * \ldots * A_k^{(\alpha_k)}$ is equal to the element (13). This completes the proof of Property 3.2 and Theorem 4.3 (b), (c). □

We shall need a few more properties of the group $G(MM_k)$.

Lemma 4.12 *Let elements $x(\vec{i}, w)$, $w \in W \cup W_0$, from $G(MM_k)$ be defined as in the proof of Lemma 4.11. Let $y \in L_1 \cup L_2$, $w \in W \cup W_0$. Then $x(\vec{i}, u)^y$ is a product of one or several elements of the form $x(\vec{i}', w')$ such that every letter a_j occurs in w' at least as many times as in w (in particular if for some $R > 0$, w belongs to the ideal V_R defined in Lemma 3.7, then $w' \in V_R$.*

Proof Indeed it is easy to check that for every $i \in \{1, 2, 3\}^{\{1,\ldots,k\}}$, $x(\vec{i}, w)$ satisfies the same equalities as elements $z(\vec{i}, w)$ from the definition of automorphism of $\overset{\diamond}{G}$ with z replaced by x everywhere.

Then the statement of the lemma for $y \in \cup_{j \geq 1} M_j$, follows from the way $x(\vec{i}, u)$ are constructed. For $y \in M_0$, one needs to use (G2), (G5) c), and (G6). □

Lemma 4.13 *The normal subgroup T of $G = G(MM_k)$ generated by the elements $x(u), u \in U$ is the direct product of cyclic subgroups generated by the elements*

$$x\left(\vec{i}, w\right), \vec{i} \in \{1, 2, 3\}^{\{1,\ldots,k\}}, w \in W \cup W_0.$$

Proof By Lemma 4.12 elements $x(\vec{i}, w)$ span T. We defined elements $x(\vec{i}, w), w \in W \cup W_0$ in such a way that they are pre-images of the corresponding elements $z(\vec{1}, w)$ in $\overset{\diamond}{G}$ under γ. Thus the elements $x(\vec{i}, w), \vec{i} \in \{1, 2, 3\}^{\{1,\ldots,k\}}, w \in W \cup W_0$ are linearly independent since their images under γ are linearly independent in T_1 (here we temporarily view T, T_1 as vector spaces over the field with p elements). □

The proof of Lemma 4.12 actually gives the following two facts (the proof of the first of them also employs Lemma 3.15 (a)).

Lemma 4.14 *If v is a word in a- and A-letters (i.e. over $L_1 \cup L_2$), $u \in U$, then $x(\vec{i}, u)^v$ is a product in G of elements $x(\vec{j}, w)$ as in Lemma 4.12 where the length of each w does not exceed $|v| + O(1)$ hence the total number of different $x(\vec{j}, w)$ occurring in this product is polynomial in $|v|$ of degree at most k.*

Lemma 4.15 *Let $v, w \in W$. Suppose $v = w$ is a Minsky relation of the semigroup $S(MM_k)$. Then the corresponding relation from (G8) has the form $x(\vec{1}, v)(x(\vec{1}, w))^{-1} = 1$. The conjugate of $x(\vec{1}, v)x(\vec{1}, w)^{-1}$ by an element $g \in \langle H_1, H_2\rangle$ is a product*

$$\prod_m (x(\vec{i}, v_m)x(\vec{i}, w_m)^{-1}),$$

where for each m, $v_m = vu_m, w_m = wu_m$ for some word u_m whose length does not exceed the length of g.

Remark 4.16 Instead of semigroup $S(MM_k)$ we could use the semigroup $\tilde{S}(MM_k)$ and construct a group $\tilde{G}(MM_k)$ which is a quotient of $G(MM_k)$ by the subgroup spanned by $x(\vec{i}, w)$ where $w \in W_0 \cup W$ contains q_0. It is easy to see that the construction of $\overset{\diamond}{G}(MM_k)$, Lemmas 4.4, 4.5, 4.8–4.15 (with notation modified in the appropriate way by "killing" q_0) and Theorem 4.18 hold for the group $\tilde{G}(MM_k)$ as well.

Moreover let $\bar{S}$ be any homomorphic image of S obtained by adding defining relations to $S(MM_k)$. Let $\breve{\bar{S}}$ be the semigroup given by the same presentation as $\bar{S}$ excluding the Minsky relations. Suppose that Properties (Q1), (Q2) from Lemma 3.15 hold in $\breve{\bar{S}}$. Then we can replace $S(MM_k)$ by $\bar{S}$, and $\breve{S}$ by $\breve{\bar{S}}$ in the construction of $G(MM_k)$. The resulting group will satisfy all the lemmas and the theorem mentioned in the previous paragraph (with $S(MM_k)$ replaced by $\bar{S}$, and $\breve{S}$ replaced by $\breve{\bar{S}}$). To verify this statement, one just needs to routinely check the proofs of these lemmas and the theorem.

4.1.1 A finitely presented solvable group with undecidable word problem

By Theorem 2.7, there exists a 2-glass Minsky machine which computes a non-recursive partial function. The corresponding group $G(MM_k)$ has undecidable word problem and belongs to the variety $\mathcal{A}_p^2\mathcal{A} \cap \mathcal{ZN}_3\mathcal{A}$ by Theorem 4.3. Hence we obtain the following:

Theorem 4.17 See ([28]). *There exists a finitely presented group with undecidable word problem that belongs to the variety* $\mathcal{A}_p^2\mathcal{A} \cap \mathcal{ZN}_3\mathcal{A}$.

The proof of this theorem presented in this paper is simpler than the original proof in [28].

4.1.2 Residually finite finitely presented groups

Theorem 4.18 *If a Minsky machine* MM_k *is sym-universally halting then the group* $G(MM_k)$ *is residually finite. The word problem in* $G(MM_k)$ *and the configuration equivalence problem for* MM_k *are polynomially reducible to each other.*

Proof Let MM_k be a sym-universally halting Minsky machine. Let $g \neq 1 \in G(MM_k)$. We use the notation from the definition of $G(MM_k)$. There exists a natural homomorphism ζ from $G(MM_k)$ to the metabelian group $H_1^{H_2} \rtimes H_2$ with kernel T. Since every finitely generated metabelian group is residually finite, we can assume that $\zeta(g) = 1$. Hence $g \in T$. By Lemma 4.13, g is a product of elements of the form

$$x(\vec{i}, w), \vec{i} \in \{1, 2, 3\}^{\{1,\ldots,k\}}, u \in W \cup W_0. \tag{14}$$

Hence $g = g_0 g_1$ where g_0 (resp. g_1) is a product of elements (14) with $w \in W_0$ (resp. $w \in W$). Suppose that g_1 is not 1. Let T' be the subgroup of $G(MM_k)$ generated by elements (14) with $w \in W_0$. Then T' is a normal subgroup of $G(MM_k)$ by Lemma 4.12. Let $G'(MM_k) = G(MM_k)/T'$. This group is a semidirect product of T/T' and the metabelian group $H_1^{H_2} \rtimes H_1$. Let D be the sum of lengths of words $w \in W$ that appear in the factors of g_1. Let Y_D be the set of all words in $\check{S}$ where at least one a-letter appears at least D times, and 0. Then Y_D is an ideal in $\check{S}$, and the image of the set of elements (14) with $w \in Y_D$ in $G'(MM_k)$ form a normal subgroup N of $G'(MM_k)$ of finite index (because T is an Abelian group of finite exponent p). That normal subgroup does not contain g by Theorem 4.3 (c). Then $G'(MM_k)/N$ is a semidirect product of a finite group and the metabelian group $H_1^{H_2} \rtimes H_2$. Hence $G'(MM_k)/N$ is residually finite and g can be separated from 1 by a homomorphism from $G(MM_k)$ onto a finite group.

Finally suppose that $g_1 = 1$. Let $w_1, \ldots, w_l$ be the elements from W_0 that appear in the representation of g as a product of elements (14). Let E be the set of words that is equal to one of the w_j in $S(MM_k)$. Since MM_k is sym-universally halting, E is finite. Let D be the maximal length of a word in E. Let, as above, Y_D be the ideal in $\check{S}$ consisting of 0 and all elements where one of the a-letters appears at least D times. Let Z_D be the set of non-zero elements of $S(MM_k)$ that are images of words from Y_D under the natural homomorphism $\check{S} \to S(MM_k)$. Then Z_D does not

contain $w_1, \ldots, w_l$. Consider the subgroup F of T spanned by all elements (14) with $w \in Z_D \cup Y_D$. From Lemma 4.12, it follows that F is a normal subgroup of $G(MM_k)$ of finite index in T. Since Z_D does not contain $w_1, \ldots, w_l$, the subgroup F does not contain g. The factor-group $G(MM_k)/F$ is a semidirect product of a finite group and the metabelian group $H_1^{H_2} \rtimes H_2$, and we can complete the proof as above.

To prove that the configuration equivalence problem in MM_k polynomially reduces to the word problem in $G(MM_k)$ we notice that the length of a word $w(c)$ corresponding to an input configuration c of MM_k in the semigroup $S(MM_k)$ is $|c| + O(1)$. If $w(c)$ is the word of length n in $S(MM_k)$, then the corresponding element in $G(MM_k)$ according to relations (G8), can be represented as a similar word $w_g(c)$ with respect to the $*$ operation. Every time when we evaluate the $*$ operation, we rewrite the word as a product of $x(q_j A_0)^u$, $u = u_1 u_2 \ldots u_k$, $u_i = a_i^{k_i} A_i$ or $u_i = a_i^{k_i}$, $j = 0, \ldots, k$ where u is a subword of $w(c)$. Using commutativity relations from Lemma 4.5, we can collect all $x(q_j A_0)^u$ with the same u. Elements $x(q_j A_0)^u$ have order p. Therefore, we have a product of conjugates $(x(q_j A^0)^u)^r$, where $1 \le r \le p-1$, and $k_1 + k_2 + k_3 \le n$. The number of such elements is at most $O(n^3)$, each of them can be written as a group word of length at most $2n + 1$. Therefore for $w(c)$ of length n, the corresponding element in $G(MM_k)$ can be represented as a word of length at most $O(n^4)$. Now two configurations c and c' of MM_k are equivalent if and only if $w(c) = w(c')$ in $S(MM_k)$ and if and only if $w_g(c) = w_g(c')$ in $G(MM_k)$ by Properties 3.1 and 3.2. Thus checking whether $c \equiv_{MM_k} c'$ can be done in polynomial time in terms of $|c| + |c'|$ with using the oracle responsible for the word problem in $G(MM_k)$ only once.

To get a polynomial reduction in the other direction we consider any element g of represented by a word w of length $\le n$ in $G(MM_k)$. We need to check if $g = 1$. First check if g is in the normal subgroup T. By construction $G(MM_k)$ is the semidirect product of T and a finitely generated metabelian group (generated by a- and A-letters). Since metabelian groups embed into finite direct products of linear groups over fields [62], the membership $g \in T$ can be checked in polynomial time. If the answer is "no", then $g \ne 1$.

Now suppose that g is in T, then we represent w in $G(MM_k)$ as a product of a fewer than n conjugates $x(\vec{i}, u)^v$ where $u \in U$, and v is a word in a-letters and A-letters whose length is bounded by n. By Lemmas 4.14 and 4.11 then w is a product in $G(MM_k)$ of elements of the form (12) and (13) whose number is bounded by a polynomial in n and whose lengths (in terms of the operation $*$) are bounded by n. Since different words of that form are linearly independent (by Lemma 4.11) in order to check whether $g = 1$, we need to verify equalities of words of the form (12) and (13) which by Theorem 4.3 is equivalent to verifying equalities of corresponding words in $S(MM_k)$, which, in turn, reduces to verifying equivalence of the corresponding configurations of MM_k by Properties 3.1 and 3.2 which hold for $S(MM_k)$. Thus the word problem in $G(MM_k)$ polynomially reduces to the configuration equivalence problem for MM_k. □

Remark 4.19 Note that if instead of the semigroup $S(MM_k)$ we could start with any semigroup $\bar{S}$ satisfying Properties (Q1), (Q2) of Lemma 3.15. By Remark 4.16, the resulting group $\bar{G}$ satisfies all the properties mentioned in that remark. The proof of Theorem 4.18 shows that if in $\bar{S}$ every non-zero element has finitely many divisors, then the group $\bar{G}$ is residually finite. Moreover if an element $w \in W_0$ has finite number

of divisors in $\bar{S}$, then there exists a homomorphism ϕ from $\bar{G}$ to a finite group with $\phi(x(\vec{1}, w)) \neq 1$.

Lemma 4.20 *Let $d(n)$ be the Dehn function of $G(MM_k)$ and $t(n)$ be the time function of MM_k. Then*

$$t(n) \preccurlyeq d(n^4)). \tag{15}$$

Proof Let $\breve{G}$ be the group given by the presentation of $G(MM_k)$ except the defining relations (G8). Let g be an element of $\breve{G}$ which is equal to 1 in $G(MM_k)$. Let w be a word in generators of $\breve{G}$ that represents g in $\breve{G}$, $|w| = n$. Then w is equal in $\breve{G}$ to a product Π of conjugates of relators (G8). This representation can be obtained as follows. Consider a minimal area van Kampen diagram Δ over the presentation of $G(MM_k)$ with boundary label w. We can read off of this diagram a representation of w as a product of conjugates of all relations (G1)–(G8). Now remove from that product all conjugates of relators (G1)–(G7). The remaining product is Π.

Let $t(n)$ be the time function of MM_k. Let c be an accepted configuration of MM_k such that $|c| = n$ and the length of the computation connecting c with the accept configuration s_0 of MM_k is $t(n)$. Let $w(c)$ and $w_g(c)$ be the corresponding elements in $S(MM_k)$ and $\breve{G}$ respectively. Let $\breve{T}$ be the normal subgroup of $\breve{G}$ generated (as a normal subgroup) by the x-letters. Since the configuration c is accepted by MM_k, we have that $w_g(c)^{-1} w_g(s_0) = 1$ modulo the relations (G8). Therefore $w_g(c)^{-1} w_g(s_0)$ is a product Π of conjugates of the relations (G8). Since the length of $w_g(c)$ does not exceed $\ell|c|^4$ for some uniform constant ℓ (see the proof of Theorem 4.18), the number of factors in the product Π does not exceed $d(\ell n^4)$.

By Lemma 4.15, $w_g(c) = x(\vec{1}, u_c)$, $w_g(c') = x(\vec{1}, u_{c'})$ for some words u_c, u'_c in the generators of $S(MM_k)$, and we can rewrite the product Π and obtain a product Π' of elements of the form $x(\vec{i}, u)x(\vec{i}, v)^{-1}$ where v is obtained from u by applying a Minsky relator of $S(MM_k)$ once. This product Π' is also equal to $x(\vec{1}, u_c)x(\vec{1}, u_{c'})^{-1}$. Since elements $x(\vec{i}, u)$, $u \in W \cup W_0$ form a basis of T viewed as a vector space of the field with p elements (Lemma 4.13), this implies that there exists an accepting computation for the configuration c of length at most $d(\ell n^4)$ and inequality (15) follows. □

Theorem 4.21 *For every recursive set of natural numbers X and every recursive function $g(n)$ there exists a finitely presented residually finite solvable of class 3 group G such that the word problem in G is as hard as the membership problem in X and polynomially reduces to it; the Dehn function G is bigger than $g(n)$.*

Proof By Theorems 2.4 and 2.7 there exists a sym-universally halting 2-glass Minsky machine MM_2 whose configuration equivalence problem polynomially reduces to the membership problem for X. By Lemma 3.5, the time complexity of the problem of recognizing equality to 0 in the semigroup $S(MM_2)$ is as large as $f(n)$ and the word problem in $S(MM_2)$ polynomially reduces to the membership problem for X. The first statement now follows from Theorems 4.18 and 3.9. For the Dehn function part of the theorem we use Lemma 4.20. Thus we just need to modify the Minsky machine MM_4 so that the new machine MM_m, $m > 4$, has the same complexity of the configuration equivalence problem but time function larger than $g(n^4)$.

This can be done in the following straightforward way. After the machine MM_4 is supposed to stop, we make MM_m compute a recursive function that is greater than $g(n^4)$, then stops. We leave it as an exercise for the reader to determine the exact value of m and the program of MM_m. □

4.2 A residually finite finitely presented group with large depth function

Theorem 4.22 *For every recursive function f and a recursive set X of natural numbers, one can construct two residually finite finitely presented solvable of class 3 groups G_1, G_2. Both groups have depth functions greater than f. The group G_1 has word problem as hard as the membership problem for X. The group G_2 has the word problem decidable in polynomial time.*

Proof Consider the Minsky machine MM_4 constructed in the proof of Theorem 3.12, the semigroup $\tilde{S}(MM_k)$ and the corresponding group $\tilde{G}(MM_4)$ (it is obtained from $G(MM_4)$ by imposing the relation $x_u = 1$ for every $u \in U$ containing q_0). Let us prove that it is residually finite. Take an element $g \in G(MM_4)$ such that $g \neq 1$. As in the proof of Theorem 4.18 we can assume $g \in T$. By Lemma 4.13, g is a product of elements of the form 14. Hence $g = g_0 g_1$ where g_0 (resp. g_1) is a product of elements (14) with $u \in W_0$ (resp. W). Suppose that g_1 is not 1. Let T' be the subgroup of $\tilde{G}(MM_4)$ generated by elements (14) with $w \in W_0$. Then T' is a normal subgroup of $\tilde{G}(MM_4)$ by Lemma 4.12 (and Remark 4.16). Let $\tilde{G}'(MM_4) = \tilde{G}(MM_4)/T'$. This group is a semidirect product of T/T' and the metabelian group $H_1^{H_2} \rtimes H_1$. Let D be the sum of lengths of words $u \in W$ that appear in the factors of g_1. Let Y_D be the set of all words in $\check{S}$ where at least one a-letter appears at least D times, and 0. Then Y_D is an ideal in $\check{S}$, and the image of the set of elements (14) with $w \in Y_D$ in $\tilde{G}'(MM_4)$ form a normal subgroup R of $G'(M_N)$ of finite index (because T is an Abelian group of finite exponent p). That normal subgroup does not contain g by Theorem 4.3 (c) (and Remark 4.16. Then $\tilde{G}'(MM_4)/R$ is a semidirect product of a finite group and the metabelian group $H_1^{H_2} \rtimes H_2$. Hence $\tilde{G}'(MM_4)/R$ is residually finite and g can be separated from 1 by a homomorphism from $\tilde{G}(MM_4)$ onto a finite group.

Thus we may assume that $g_1 = 1$. Let $w_1, \ldots, w_l$ be the elements from W_0 that appear in the representation of g as a product of elements (14). Let E be the set of elements of $\hat{S}$ (i.e., words modulo the commutativity relations) which are equal to one of the w_j in $S(MM_4)$. Note that every word in E starts with a q-letter by definition of elements (14).

For each $w \in E$ let $D(w)$ be as in the proof of Theorem 3.12. Let D be the maximum of these numbers $D(w)$ and, as in the proof of Theorem 3.12, let $\bar{S}$ be the semigroup $\tilde{S}(MM_4)$ with additional defining relations $a_3^D = a_3^{2D}$, $a_4^D = a_4^{2D}$. As we mentioned in the proof of Theorem 3.12, every non-zero element of $\bar{S}$ has finitely many divisors. It is easy to see that $\bar{S}$ satisfies Properties (Q1), (Q2) of Lemma 3.15. Therefore by Remark 4.16, we can built a group $\bar{G}$ starting with $\bar{S}$ instead of $\tilde{S}(MM_4)$ and all the statements mentioned in Remark 4.16 remain true for $\bar{G}$. The natural homomorphism $\delta\colon \tilde{S}(MM_4) \to \bar{S}$ extends to a homomorphism $\bar{\delta}\colon \tilde{G}(MM_k) \to \bar{G}$. By Theorem 4.3 and Lemma 4.13 the images under $\bar{\delta}$ of all elements (14) with $w \in \{w_1, \ldots, w_l\}$ form

a finite linearly independent set. Hence $\bar{\delta}(g) \neq 1$ in $\bar{G}$. Since every non-zero element of $\bar{S}$ has finitely many divisors, the group $\bar{G}$ is residually finite by Remark 4.19. Hence there exists a homomorphism of $\tilde{G}(MM_4)$ onto a finite groups separating g from 1. Thus $\tilde{G}(MM_4)$ is indeed residually finite.

The fact that $\rho_G(n) \geq f(n)$ is proved the same way as in the proof of Theorem 3.12 (one only needs to replace the product by operation * everywhere in that proof). □

5 Applications

In this section we present two applications of our results and methods. One application concerns the universal theory of finite solvable groups of a given class, the second application concerns with the membership problem in pro-finitely closed subgroups of residually finite groups.

5.1 Universal theories of sets of finite groups

In this section we will prove the following result. For the class of all finite groups in was proved by Slobodskoi [60] (the idea of Slobodskoi's proof came from Gurevich's paper [23] where the same result was proved for semigroups, see also [58]).

Theorem 5.1 *The universal theories of the class of finite groups from $\mathcal{A}_p^2\mathcal{A} \cap \mathcal{ZN}_5\mathcal{A}$ and the class of all periodic groups are recursively inseparable. In particular, the universal theory of any set of finite groups containing all finite solvable of class 3 groups is undecidable.*

Proof Consider a partial function $f : \mathbb{N} \to \mathbb{N}$ such that f is one-to-one and computable on its domain, and the domain is recursively enumerable but not recursive. Let MM_2 be a Minsky machine computing the function f. We can assume that for every number not in the domain of f the machine MM_2 works indefinitely long and never stops. As in the proof of Theorem 3.12, we can assume that the start command number 1 cannot be used in the middle of a computation of MM_2.

Consider the 4-glass Minsky machine MM_4 described in the proof of Theorem 3.12. Let $S'(MM_4)$ be the semigroup given by the same defining relations as $\tilde{S}(MM_4)$ except the relation $q_0 = 0$ is substituted by the relation $q_i A_3 A_4 = 0$ for every i.

Let $G'(MM_4)$ be the group corresponding to $S'(MM_4)$ in the same way $G(MM_k)$ corresponds to $S(MM_k)$. Since Properties (Q1), (Q2) of Lemma 3.15 obviously hold for $S'(MM_4)$ we can use Remark 4.16. Then $G'(MM_4)$ belongs to $\mathcal{A}_p^2\mathcal{A} \cap \mathcal{ZN}_5\mathcal{A}$ and simulates MM_4 as described in Theorem 4.3. Let R be the (finite) set of defining relations of $G'(MM_4)$. Let X be the set of numbers ϵ such that MM_4 accepts the configuration $(\epsilon, 0, 0, 0)$. Let X' be the set of numbers ϵ such that MM_4 works infinitely long starting with the configuration $(\epsilon, 0, 0, 0)$. Then X and X' are recursively inseparable by the choice of MM_2 and MM_4. For any configuration $(\epsilon, 0, 0, 0)$ of MM_4 consider the corresponding element

$$w(\epsilon) = q_1 * a_1^{(\epsilon)} * A_1 * A_2 * a_3 * A_3 * A_4.$$

Suppose $\epsilon \in X$. Then the Minsky machine MM_2 halts starting at $(\epsilon, 0, 0, 0)$. Since the function f is one-to-one, there are only finite number of computations of $\mathrm{Sym}(MM_2)$ starting at the configuration $c_\epsilon = (1; \epsilon, 0, 0, 0)$. Then as in the proof of Theorem 4.22, we can find a a homomorphism ϕ from $\tilde{G}(MM_4)$ to a finite group such that $\phi(x(\vec{1}, w)) \neq 1$.

Hence the universal formula $\& R \to x(\vec{1}, w) = 1$ does not hold in the finite group H from $\mathcal{A}_p^2\mathcal{A} \cap \mathcal{Z}\mathcal{N}_5\mathcal{A}$.

Now suppose that $\epsilon \in X'$. Consider any periodic homomorphic image H of $G'(MM_4)$. Let $\bar{t}$ be the image of $t \in G'(M_n)$ in H. Then there exists a number D such that for every element $x \in \bar{T}$,

$$x * \bar{a}_3^{(D)} = x * \bar{a}_3^{(2D)}. \tag{16}$$

Since MM_4 works infinitely long starting at the configuration $(\epsilon, 0, 1, 0)$, by Theorem 4.3 the following equality is true for some i, k_1, k_2:

$$w(\epsilon) = \bar{x}(\vec{1}, q_i A_0) * \bar{a}_1^{(k_1)} * \bar{a}_2^{(k_2)} * \bar{a}_3^{(D)} * \bar{A}_1 * \bar{A}_2 * \bar{A}_3 * \bar{A}_4.$$

Then by (16) and Theorem 4.3

$$\begin{aligned}\bar{w}(\epsilon) &= \bar{x}_{\vec{1}, q_i A_0} * \bar{a}_1^{(k_1)} * \bar{a}_2^{(k_2)} * \bar{a}_3^{(2D)} * \bar{a}_4^{(D)} * \bar{A}_1 * \bar{A}_2 * \bar{A}_3 * \bar{A}_4 \\ &= \bar{x}_{\vec{1}, q_i A_0} * \bar{a}_1^{(k_1)} * \bar{a}_2^{(k_2)} * \bar{a}_3^{(D)} * \bar{a}_4^{(D)} * \bar{A}_1 * \bar{A}_2 * \bar{A}_3 * \bar{A}_4 \\ &= \bar{x}_{\vec{1}, q_i A_0} * \bar{a}_1^{(k_1)} * \bar{a}_2^{(k_2)} * \bar{A}_1 * \bar{A}_2 * \bar{A}_3 * \bar{A}_4 = 1.\end{aligned}$$

since $q_i A_3 A_4 = 0$ in $S'(M_n)$. Hence the universal formula $\& R \to w(\epsilon) = 1$ holds in H.

Thus the set of universal formulas $\& R \to w(\epsilon) = 1$ that do not hold in some finite group from $\mathcal{A}_p^2\mathcal{A} \cap \mathcal{Z}\mathcal{N}_5\mathcal{A}$ and the set of such formulas which hold in every periodic group are recursively inseparable. □

Remark 5.2 Note that the universal theory of finite metabelian groups is decidable because every finitely generated metabelian group is residually finite (the connection is explained in [32]). On the other hand, the universal theory of all finite nilpotent groups is undecidable [30]. The description of all (finitely based) varieties of groups where the universal theory of finite groups is decidable is currently out of reach. In fact our Theorem 5.1 gives the first example of a proper variety of groups where the universal theory of finite groups is undecidable. From Zelmanov's solution of the restricted Burnside problem [65,66], it immediately follows that the universal theory of finite groups in every finitely based periodic variety of groups is decidable. That result and simulations of Minsky machines in semigroups (as in Sect. 3) were used by the third author [56] to obtain a complete description of all finitely based varieties of semigroups where finite semigroups have decidable universal theory. For more information on that problem, see [32].

5.2 Distortion of pro-finitely closed subgroups of finitely presented groups

Let G be a group generated by a finite set X, $H \leq G$ be a subgroup generated by a finite set Y. Recall that the distortion function $f_{H,G}(n)$ is defined as the minimal number k such that every element of H represented as a word w of length $\leq n$ in the alphabet X can be represented as a word of length $\leq k$ in the alphabet Y [17]. Distortion functions with respect to two different sets of generators for the same group are equivalent. By [17] in a group G with decidable word problem, the distortion function $f_{G,H}$ is recursive if and only if the membership problem in H is decidable.

Recall that H is *closed in the pro-finite topology* of G if H is the intersection of some subgroups of G of finite index. If G is finitely presented and H is closed in the pro-finite topology of G, then there exists a McKinsey-type algorithm $A(G, H)$ solving the membership problem for H (and thus the $f_{G,H}$ is recursive). For every word w in the alphabet X, the "yes" part $A_{\text{yes}}(G, H)$ of the algorithm lists all words in Y, rewrites them as words in X, and then applies relations of G to check whether one of these words is equal to w. The "no" part $A_{\text{no}}(G, H)$ of the algorithm lists all homomorphisms ϕ of G into finite groups and checks whether $\phi(w) \notin \phi(H)$. As in Sect. 1.2, one can asks what is the complexity of the "yes" and "no" parts of that algorithm, in particular, and of the membership problem for H in general.

The time complexity of $A_{\text{yes}}(G, H)$ can be estimated in terms of the distortion function $f_{G,H}(n)$ and the time complexity of $A_{\text{no}}(G, H)$ can be estimated estimated in terms of the *relative depth function* $\rho_{G,H}(n)$ which is defined as the minimal number r such that for every word w of length $\leq n$ in X which does not represent an element of H there exists a homomorphism ϕ from G to a finite group of order $\leq r$ such that $\phi(w) \notin \phi(H)$.

As for the word problem in residually finite finitely presented groups (discussed above), there were no examples of finitely generated subgroups of finitely presented groups that are closed in the pro-finite topology but have "arbitrary bad" distortion or "arbitrary bad" relative depth function.

Mikhailova's construction [42] shows that finitely generated subgroups of the residually finite group $F_2 \times F_2$ (here F_2 is a free group of rank 2) could be very distorted. In fact the set of possible distortion functions of subgroups of $F_2 \times F_2$ coincides, up to a natural equivalence, with the set of Dehn functions of finitely presented groups [47]. Finitely generated subgroups of $F_2 \times F_2$ are *equalizers* of pairs of homomorphisms $\phi\colon F_k \to G, \psi\colon F_n \to G$ (where F_k, F_n are some subgroups of F_2), i.e. the subgroups of the form $\{(x, y) \in F_k \times F_n \mid \phi(x) = \psi(y)\}$ (see, for example, [52] or[5]). The equalizer subgroup is finitely generated if and only if G is finitely presented [5].

It is easy to prove (see Lemma 5.3 below) that if G is residually finite, then the equalizer is closed in the pro-finite topology of $F_2 \times F_2$. In fact we have the following more general statement:

Lemma 5.3 *Let $\mathcal{P}$ be a class of finite groups closed under direct products and subgroups. Let G be a finitely generated group, let N be a normal subgroup of G, and let ϕ, ψ be two homomorphisms $G \to G/N$. If G/N is residually $\mathcal{P}$, then the equalizer $E(\phi, \psi) = \{(g, h \in G \times G \mid \phi(g) = \psi(h)\}$ is closed in the pro-$\mathcal{P}$ topology on $G \times G$.*

Proof Suppose $(u, v) \in G \times G$ but $(u, v) \notin E(\phi, \psi)$, so $\phi(u) \neq \psi(v)$. Since G/N is residually $\mathcal{P}$ there is a homomorphism $\eta : G/N \to K$ onto a finite group $K \in \mathcal{P}$ such that $\eta\phi(u) \neq \eta\psi(v)$ in K. Therefore the image of the pair (u, v) under $(\eta\phi, \eta\psi)$ is not in the image of the subgroup $E(\phi, \psi)$ in $K \times K$. Hence the subgroup $E(\phi, \psi)$ is closed in the pro-$\mathcal{P}$ topology on $G \times G$. □

Lemma 5.3 and Theorems 4.22 and 4.21 immediately imply

Corollary 5.4 *For every recursive function $f(n)$ there exists a finitely generated subgroup $H \leq F_2 \times F_2$ that is closed in the pro-finite topology of $F_2 \times F_2$ and whose distortion function $f_{F_2 \times F_2, H}$, the relative depth function, and the time complexity of the membership problem are at least $f(n)$.*

Remark 5.5 Since the groups we construct are solvable of class 3, a similar corollary is true with F_2 replaced by the free solvable group of class 3 of finite rank (although the rank is not necessarily 2 because not every free solvable group of class 3 embeds into a 2-generated group that is solvable of class 3).

Acknowledgements The authors are grateful to Jean-Camille Birget and Friedrich Otto for pointing to the references [13], to Ben Steinberg for pointing to the reference [34],to Rostislav Grigorchuk for pointing to the references [15,21] and to Tim Riley for pointing to the references [19,20]. We are also grateful to Markus Lohrey and Ralph Strebel for their comments. We are especially grateful to the anonymous referees whose numerous suggestions helped us improve the paper.

References

1. Agol, I.: The virtual Hacken conjecture (with an appendix by I. Agol, D. Groves and J. Manning), arXiv:1204.2810, (2012)
2. Baumslag, G.: A non-cyclic one-relator group all of whose finite factor groups are cyclic. J. Aust. Math. Soc. **10**, 497–498 (1969)
3. Baumslag, G.: Subgroups of finitely presented metabelian groups. J. Aust. Math. Soc. Ser. A **16**(1), 98–110 (1973)
4. Baumslag, G., Miller III, C.F., Short, H.: Isoperimetric inequalities and the homology of groups. Invent. Math. **113**(3), 531–560 (1993)
5. Baumslag, G., Roseblade, J.E.: Subgroups of direct products of free groups. J. Lond. Math. Soc. **30**, 44–52 (1984)
6. Borisov, A., Sapir, M.: Polynomial maps over finite fields and residual finiteness of mapping tori of group endomorphisms. Invent. Math. **160**(2), 341–356 (2005)
7. Borisov, A., Sapir, M.: Polynomial maps over p-adics and residual properties of mapping tori of group endomorphisms. Int. Math. Res. Not. IMRN **16**, 3002–3015 (2009)
8. Birget, J.-C.: Infinite string rewriting systems and complexity. J. Symb. Comput. **25**(6), 759–793 (1998)
9. Bou-Rabee, K.: Quantifying residual finiteness. J. Algebra **323**, 729–737 (2010)
10. Brady, N., Dison, W., Riley, T.: Hyperbolic hydra. Groups Geom. Dyn. **7**(4), 961–976 (2013)
11. Bridson, M., Haefliger, A.: Metric Spaces of Non-Positive Curvature. Springer, Berlin (1999)
12. Cohen, D.E.: Combinatorial Group Theory: A Topological Approach. London Mathematical Society Student Texts, 14. Cambridge University Press, Cambridge (1989)
13. Davis, M.D.: A note on universal Turing machines. Automata studies, Annals of Mathematics Studies, no. 34, pp. 167–175. Princeton University Press, Princeton (1956)

14. Dison, W., Riley, T.: Hydra groups. Comment. Math. Helv. **88**(3), 507–540 (2013)
15. Dyson, V.H.: A family of groups with nice word problems. Collection of articles dedicated to the memory of Hanna Neumann, VIII. J. Aust. Math. Soc. **17**, 414–425 (1974)
16. Ershov, M.: Golod–Shafarevich groups: a survey. Int. J. Algebra Comput. **22**(5), 1230001 (2012)
17. Farb, B.: The extrinsic geometry of subgroups and the generalized word problem. Proc. Lond. Math. Soc. **68**(3), 577–593 (1994)
18. Gersten, S.M.: Dehn functions and l1-norms of finite presentations. Algorithms and Classification in Combinatorial Group Theory, pp. 195–225. Springer, Berlin (1992)
19. Gersten, S.M.: Isoperimetric and isodiametric functions of finite presentations. Geometric Group Theory, vol. 1 (Sussex, 1991), pp. 79–96, London Math. Soc. Lecture Note Ser., 181, Cambridge University Press, Cambridge (1993)
20. Gersten, S.M., Riley, T.R.: Some duality conjectures for finite graphs and their group theoretic consequences. Proc. Edinb. Math. Soc. **48**(2), 389–421 (2005)
21. Grigorchuk, R.: Groups with intermediate growth functions and their applications, Doctor's Thesis (Russian), Moscow Steklov Mathematical Institute (1985)
22. Golubov, E.A.: Finite separability in semigroups. Dokl. Akad. Nauk SSSR **189**, 20–22 (1969)
23. Gurevich, Y.S.: The problem of equality of words for certain classes of semigroups. Algebra i Log. Sem. **5**(5), 25–35 (1966)
24. Higman, G.: Subgroups of finitely presented groups. Proc. R. Soc. Ser. A **262**, 455–475 (1961)
25. Hsu, T., Wise, D.: Cubulating graphs of free groups with cyclic edge groups. Amer. J. Math. **132**(5), 1153–1188 (2010)
26. Kassabov, M., Matucci, F.: Bounding the residual finiteness of free groups. Preprint, arXiv
27. Kassabov, M., Nikolov, N.: Generation of polycyclic groups. J. Group Theory **12**(4), 567–577 (2009)
28. Kharlampovich, O.G.: Finitely presented solvable group with unsolvable word problem. Sov. Math. Izvest. **45**(4), 852–873 (1981)
29. Kharlampovich, O.G.: The word problem for groups and Lie algebras, Doctor's Thesis (Russian), Moscow Steklov Mathematical Institute (1990)
30. Kharlampovich, O.G.: The universal theory of the class of finite nilpotent groups is undecidable. Mat. Zametki **33**(4), 499–516 (1983)
31. Kharlampovich, O.G., Sapir, M.V.: A non-residually finite, relatively finitely presented group in the variety $\mathfrak{N}_2\mathfrak{A}$. Combinatorial and Geometric Group Theory (Edinburgh, 1993), pp. 184–189, London Math. Soc. Lecture Note Ser., 204, Cambridge Universiy Press, Cambridge (1995)
32. Kharlampovich, O., Sapir, M.: Algorithmic problem in varieties. Int. J. Algebra Comput. **5**(4–5), 379–602 (1995)
33. Kourovskaja tetrad' (Unsolved Problems in Group Theory), 5th edn. Novosibirsk, (1976)
34. Lipton, R.J., Zalcstein, Y.: Word problems solvable in logspace. J. Assoc. Comput. Mach. **24**, 522–526 (1977)
35. Malcev, A.I.: Algorithms and Recursive Functions. Nauka, Moscow (1965)
36. Malcev, A.I.: On Homomorphisms onto finite groups (Russian). Uchen. Zap. Ivanovskogo Gos. Ped. Inst. **18** (1958), pp. 49–60. English translation in: Amer. Math. Soc. Transl. Ser. 2, **119**, pp. 67–79 (1983)
37. McKenzie, R., Thompson, R.J.: An elementary construction of unsolvable word problems in group theory. Word problems: decision problems and the Burnside problem in group theory (Conf., Univ. California, Irvine, Calif. 1969; dedicated to Hanna Neumann), Studies in Logic and the Foundations of Math., 71, p. 457478. Amsterdam (1973)
38. Meskin, S.: A finitely generated residually finite group with an unsolvable word problem. Proc. Am. Math. Soc. **43**(1), 8–10 (1974)
39. Madlener, K., Otto, F.: Pseudonatural algorithms for the word problem for finitely presented monoids and groups. J. Symb. Comput. **1**(4), 383–418 (1985)
40. McKinsey, J.: The decision problem for some classes of sentences without quantifiers. J. Symb. Log. **8**, 61–76 (1973)
41. Miasnikov, A., Ushakov, A., Won, D.: The word problem in Baumslag group is polynomial time decidable. J. Algebra **345**, 324–342 (2011)
42. Mikhailova, K.A.: The occurrence problem for direct products of groups. Dokl. Akad. Nauk SSSR **119**, 1103–1105 (1958)
43. Neumann, H.: Varieties of Groups. Springer, Berlin (1967)

44. Nikolov, N., Segal, D.: Finite index subgroups in pro-finite groups. C. R. Math. Acad. Sci. Paris **337**(5), 303–308 (2003)
45. Ollivier, Y., Wise, D.T.: Cubulating random groups at density less than 1/6. Trans. Am. Math. Soc. **363**(9), 4701–4733 (2011)
46. Olshanskii, AYu.: Almost every group is hyperbolic. Int. J. Algebra Comput. **2**(1), 1–17 (1992)
47. Olshanskii, A., Sapir, M.: Length and area functions on groups and quasi-isometric Higman embeddings. Int. J. Algebra Comput. **11**, 137–170 (2001)
48. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley Publishing Company, Reading (1994)
49. Pueschel, K.: Hydra group doubles are not residually finite, arXiv:1507.02554
50. Platonov, A.N.: An isoperametric function of the Baumslag–Gersten group. Vestnik Moskov. Univ. Ser. I Mat. Mekh. 2004, no. 3, pp. 12–17, translation in Moscow Univ. Math. Bull. 59 (2004), no. 3, p. 1217 (2005)
51. Remeslennikov, V.: Studies on infinite solvable and finitely approximable groups. Mat. Zametki **17**(5), 819–824 (1975)
52. Remak, R.: Uber der Zerlegung der endlichen Gruppen in direkte unzerlegbare Faktoren. J. Reine Angew. Math. **139**, 293308 (1911)
53. Rips, E.: Subgroups of small cancellation groups. Bull. Lond. Math. Soc. **14**, 45–47 (1982)
54. Rotman, J.J.: An Introduction to the Theory of Groups, 4th edn. Graduate Texts in Mathematics, 148. Springer, New York (1995)
55. Sapir, M.: Algorithmic problems in varieties of semigroups. Algebra i Logika **27**(4), 440–463 (1988)
56. Sapir, M.: Weak word problem for finite semigroups. Monoids and Semigroups with Applications (Berkeley, CA, 1989), pp. 206–219. World Science Publisher, River Edge (1991)
57. Sapir, M.: Asymptotic invariants, complexity of groups and related problems. Bull. Math. Sci. **1**(2), 277–364 (2011)
58. Sapir, M.: Minsky machines and algorithmic problems, accepted in Essays Dedicated to Yuri Gurevich on the Occasion of His 75th Birthday, LNCS. Springer, Berlin (2015)
59. Sapir, M., Birget, J.C., Rips, E.: Isoperimetric and isodiametric functions of groups. Ann. Math. **156**(2), 345–466 (2002)
60. Slobodskoi, A.M.: Undecidability of the universal theory of finite groups. Algebra Log. **20**(2), 207–230 (1981)
61. Waack, St: On the parallel complexity of linear groups. RAIRO Inform. Theor. Appl. **25**, 323–354 (1991)
62. Wehrfritz, B.A.F.: On finitely generated soluble linear groups. Math. Z. **170**(2), 155–167 (1980)
63. Wise, D.T.: The structure of groups with a quasiconvex hierarhy. Preprint (2011)
64. Wise, D.T.: A residually finite version of Rips's construction. Bull. Lond. Math. Soc. **35**(1), 23–29 (2003)
65. Zel'manov, E.I.: The solution of the restricted Burnside problem for groups of odd exponent. Izv. Akad. Nauk. SSSR. Ser. Mat., **54**(1), 42–59 (1990). Transl. in Math. USSR-Izv. **36**(1), 41–60 (1991)
66. Zel'manov, E.I.: The solution of the restricted Burnside problem for 2-groups. Mat. Sb. **182**(4), 568–592 (1991)