The word and order problems for self-similar and automata groups

Laurent Bartholdi, Ivan Mitrofanov

Groups Geom. Dyn., DRAFT, 1–24
DOI 10.4171/GGD/??

Groups, Geometry, and Dynamics

The word and order problems for self-similar and automata groups

Laurent Bartholdi and Ivan Mitrofanov

Abstract. We prove that the word problem is undecidable in functionally recursive groups, and that the order problem is undecidable in automata groups, even under the assumption that they are contracting.

Mathematics Subject Classification (2010). 20F10, 20F65, 03D10, 68Q05, 68Q80.

Keywords. Self-similar groups, automata groups, word problem, order problem, contracting groups, Mealy machines, Minsky machines.

1. Introduction

Let A be a finite set (the *alphabet*), and consider a group G acting faithfully and "self-similarly" on the set A^* of words over A. This means that every $g \in G$ acts in the form

$$(a_1 \dots a_n)^g = a_1'(a_2 \dots a_n)^{g'} \tag{1.1}$$

for some $a_1' \in A$ and some $g' \in G$ depending only on a_1, g; we encode them as $(g', a_1') = \bar{\Phi}(a_1, g)$ for a map $\bar{\Phi}\colon A \times G \to G \times A$. If furthermore G is finitely generated (say by a finite set S of *states*, so G is a quotient $F_S \twoheadrightarrow G$ of the free group on S), then its action may be described by finite data, namely a lift $\Phi\colon A \times S \to F_S \times A$ of the restriction of $\bar{\Phi}$ to the generators of G. A finitely generated group given in this manner is called *functionally recursive* [6, §3], or *self-similar*; we call G the group *presented* by Φ, and write $G = \langle \Phi \rangle$, and we call Φ a (state-asynchronous) *transducer*. Note that we restrict ourselves to *alphabet-synchronous* transducers (see the remark in §1.5): they read and write precisely one letter at each clock tick.

Even though the map Φ completely determines the action of G, and therefore G itself, it is unclear how much of G is known from Φ. Our first result is as negative as can be:

Theorem A. *There is no algorithm that, given $\Phi\colon A \times S \to F_S \times A$ and $s \in S$, determines whether $s = 1$ in $\langle \Phi \rangle$.*

Large classes of finitely generated groups can be presented as functionally recursive ones; notably, all the "iterated monodromy groups" of Nekrashevych [20], and the automata groups mentioned in §1.1 below.

1.1. Automata groups. Assume now that G is a functionally recursive group, and that in the action (1.1) the elements g' have length at most the length of g, in the generating set S. Then, up to replacing S by $S \cup S^{-1} \cup \{1\}$, the map Φ takes the form $\Phi\colon A \times S \to S \times A$; we call it a *finite state transducer*. The group G is called an *automata group*; these form a notorious class of groups, containing all finitely generated linear groups as well as infinite torsion groups such as the "Grigorchuk group" [9] and "Gupta–Sidki groups" [12]. The Grigorchuk group is also a group of intermediate word-growth, and was used to settle the Milnor problem on group growth [10].

The action of S, and of G itself, may be conveniently described by a finite labeled graph called its *Moore diagram*. Consider the directed graph Γ with vertex set S and an edge from s to t labeled (a, b) whenever $\Phi(a, s) = (t, b)$; then the action of $s \in S$ on A^* is determined as follows: given $a_1 \dots a_n \in A^*$, find the unique path in Γ starting at s and whose first label letters read $a_1 \dots a_n$; let $b_1 \dots b_n$ be the second label letters; then $(a_1 \dots a_n)^s = b_1 \dots b_n$. See Figure 1 for the graph Γ describing the Grigorchuk group.

Every element of G (say represented by a word w of length n in S) admits a similar description, but now using a graph with vertex set S^n. The word w represents the identity in G if and only if at every vertex reachable from w all the outgoing edges have labels in $\{(a, a) \mid a \in A\}$. It follows that the word problem is decidable in G, and even belongs to LinSpace (and therefore to ExpTime); but that is about as much as is known. We consider the "order problem" (determine the order of an element), which was raised at the end of last century by Sidki [22, §§5.2-5.3, and public lectures] and by Grigorchuk, Nekrashevych and Sushchansky [11, Problem 7.2.1(a)], and was independently solved by Gillibert [8]:

Theorem B. *There is no algorithm that, given $\Phi\colon A \times S \to S \times A$ and $s \in S$, determines the order of s in $\langle \Phi \rangle$, namely the cardinality of $\langle s \rangle$.*

Worse than that, the action is uncomputable in the following sense: consider the natural extension of the action of $\langle \Phi \rangle$ to A^∞. Then we have the following variants of Theorems A and B:

Theorem A′. *There is no algorithm that, given $\Phi\colon A \times S \to F_S \times A$ and $a \in A$ and $s \in S$, determines whether a^∞ is fixed by s.*

Theorem B′. *There is no algorithm that, given $\Phi\colon A \times S \to S \times A$ and $a \in A$ and $s \in S$, determines the cardinality of the orbit of a^∞ under $\langle s \rangle$.*

Finally, the results in Theorems A and B can be expressed in a uniform framework as follows:

Theorem A″. *There is a functionally recursive group* $\langle\Phi\rangle$ *with* $\Phi\colon A\times S \to F_S\times A$ *such that* $\{s \in F_S \mid s = 1 \text{ in } \langle\Phi\rangle\}$ *is not recursive.*

Theorem B″. *There is an automata group* $\langle\Phi\rangle$ *with* $\Phi\colon A \times S \to S \times A$*, and two states* $s, t \in S$*, such that the set* $\{n \in \mathbb{N} \mid st^n$ *has finite order*$\}$ *is not recursive.*

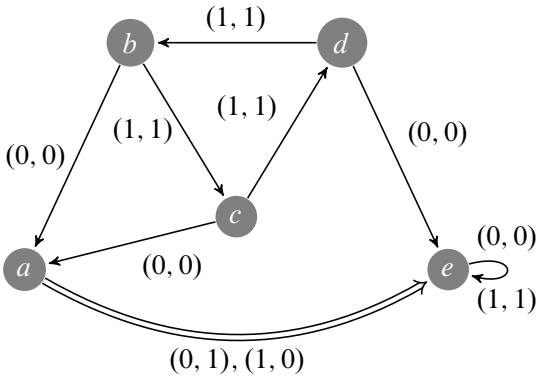


Figure 1. The transducer generating the Grigorchuk group. Here $A = \{0, 1\}$ and $S = \{a, b, c, d, e\}$.

1.2. Contracting groups. Assume now that G is a functionally recursive group, and that in the action (1.1) the elements g' are *much shorter* than g, in the generating set S, in the sense that there are constants $\lambda < 1$ and C with $|g'| \leq \lambda|g| + C$ for all $g \in G$. Then, up to replacing S by the set of all words of length $\leq C/(1 - \lambda)$, we also have $|g'| \leq |g|$; we have thus defined a subclass of automata groups, called *contracting automata groups* (see §3.4 for a more precise definition). Their word problem is decidable in LogSpace (and therefore in PolyTime). We will see, however, that the order and orbit order problems remain unsolvable in that restricted class:

Theorem C (= Theorem 3.4). *The transducers constructed in Theorems* B *and* B′ *may be assumed to generate contracting groups.*

1.3. Sketch of proofs. We encode Minsky machines in functionally recursive groups. Minsky machines (see [19, Chapter 11]) are restricted Turing machines with two tapes, which may move the tapes and sense the tapes' end but may not write on them; equivalently, they are finite state automata equipped with two counters with values in $\mathbb{N}$ that may be incremented, decremented and tested for 0.

Let **M** be a Minsky machine. We construct a functionally recursive group containing elements x, y and an element s for each state of **M**, and encode configurations of **M** in the group as follows: when the machine is in state s with counter values (m, n), we encode it by the word $sx^{2^m}y^{2^n}$. The action of the group is so devised that if the machine transitions from (s, m, n) to (s', m', n') then the action of $sx^{2^m}y^{2^n}$ induces, on a certain subtree, the action of $s'x^{2^{m'}}y^{2^{n'}}$. It then follows that the image of a prescribed ray under $sx^{2^m}y^{2^n}$ records the computational steps of **M** when started in (s, m, n), and in particular whether the machine reached a final state—the image of the ray will then contain a certain marker symbol. We construct an auxiliary element t that only acts on sequences containing this marker symbol, and then $(sx^{2^m}y^{2^n})t(sx^{2^m}y^{2^n})^{-1}$ fixes the original ray if and only if the machine never reaches the final state. Taking the commutator of that last element with an element acting only in the neighbourhood of the original ray yields a group element that is trivial if and only if the machine never reaches the final state.

It is an inherent part of the construction that sometimes the output of the transducer is longer than the input (e.g., if **M** increments its first counter, the functional recursion must replace x by x^2).

To obtain an automata group, namely a group generated by a state-synchronous transducer, we modify the construction above by having the transducer consume a power of its input word $sx^{2^m}y^{2^n}$ to produce $s'x^{2^{m'}}y^{2^{n'}}$; e.g., the incrementation of the first counter may be performed by erasing every second s and every second block of y^{2^n}'s. A transducer is constructed out of **M** in such a manner that if **M** runs forever when started in configuration $(s_*, 0, 0)$, passing through configurations (s_i, m_i, n_i) for $i = 0, 1, \ldots$, then the orbit under s_*xy of some ray (constructed out of $(s_0, s_1, \ldots)$) will be infinite so s_*xy has infinite order; while if **M** stops then s_*xy has finite order.

1.4. Tilings. Our results on functionally recursive groups and transducers may also be interpreted in terms of tilings. Let C be a finite set of *colours*, and let $T \subseteq C^{N,E,S,W}$ be a set of *Wang tiles*. A *valid tiling* is a map $t\colon \mathbb{Z}^2 \to T$ with $t(x, y)^N = t(x, y+1)^S$ and $t(x, y)^E = t(x+1, y)^W$ for all $x, y \in \mathbb{Z}^2$. Berger showed in [4] that it is undecidable to determine, given T, whether there exists a valid tiling by T. This has been improved: for $\lambda, \mu \in \{N, E, S, W\}$, call a set of tiles $\lambda\mu$-*deterministic* if for every $c, d \in C$ there exists at most one tile $u \in T$ with $u^\lambda = c$ and $u^\mu = d$, and $\lambda\mu$-*complete* if there exists precisely one tile $u \in T$ with these conditions. Lukkarila showed in [17] that the undecidability result holds even under the restriction that T is NE, NW, SE, SW-deterministic. Clearly a SW-complete tileset tiles uniquely the first quadrant for any choice of colours on the axes.

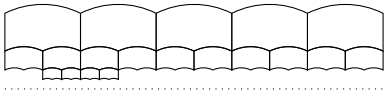
Our result on the order problem has the following translation into tilings. We consider tilings of the upper half-plane $\{(x, y) \mid y \geq 0\}$. Then the following

problem is undecidable even for SE, SW-complete tilesets:

> *"given $c \in C$, is there an integer $n \in \mathbb{N}$ such that every tiling of the upper half-plane with c^∞ on the horizontal axis is horizontally n-periodic?"* (1.2)

Indeed, given $\Phi\colon A\times S \to S\times A$, set $C = A \sqcup S$ and whenever $\Phi(a,s) = (s',a')$ build a tile with N,E,S,W-labels s',a',s,a respectively; also build tiles with N,E,S,W-labels c,d,c,d for all $(c,d) \in C^2 \setminus (S\times A)$. Then the tiling problem in (1.2) has a solution for $c \in S$ if and only if c has finite order in $\langle\Phi\rangle$.

The word problem may also be translated to a tiling problem, but now in hyperbolic space. The tileset is now $T \subseteq C^{N,E,S_1,S_2,W}$. The lattice $\mathbb{Z}^2$ is now $\Lambda := \{2^y(i+x) \mid x,y \in \mathbb{Z}\} \subset \mathbb{H}$. A tiling is a map $t\colon \Lambda \to T$ with $t(2^y(i+x))^E = t(2^y(i+x+1))^W$ and $t(2^y(i+2x))^N = t(2^{y+1}(i+x))^{S_1}$ and $t(2^y(i+2x+1))^N = t(2^{y+1}(i+x))^{S_2}$ for all $x,y \in \mathbb{Z}$. Tiles are visualized as pentagons assembling into a tiling of the hyperbolic plane, invariant under the transformations $z \mapsto z+1$ and $z \mapsto 2z$:



The following problem is undecidable even for NE, NW-complete tilesets: *"given $c \in C$, does every tiling of $\{x+iy \in \mathbb{H} \mid x \in [0,1], y \le 1\}$ with c on the edge from i to $i+1$ have identical labels on the boundary half-lines $\{x=0\}$ and $\{x=1\}$?"*.

Indeed by subdividing and inserting the empty state we may assume that the map Φ describing our functionally recursive group satisfies $\Phi(A\times S) \subseteq S^2 \times A$; then tiles are defined as above.

1.5. History. Links have been established since the beginning between undecidable problems in theoretical computer science—halting of Turing machines—and in algebra—word problems for instance. Minsky machines, because of their simplicity, have been early recognized as useful tools in this correspondence, see e.g. Gurevich's work [13] on identities in semigroups.

Automata semigroups are defined quite similarly to automata groups; one merely drops the requirement that the action be by invertible maps. Decision problems have been extensively studied within the class of automata semigroups [1, 15]. Gillibert proved in [7] that the order problem is unsolvable in that class. His proof is based on the undecidability of Wang's tiling problem [4], and harnesses Kari's solution of the nilpotency problem for cellular automata [14].

There are usually serious difficulties in converting a solution in semigroups to one in groups. In particular, the tilings at the heart of Gillibert's construction give fundamentally non-invertible transformations of A^*.

On the other hand, a direct approach to the order problem succeeded for the restricted class of "bounded automata" groups; Bondarenko, Sidki and Zapata prove in [5] that they have solvable order problem.

The general context considered by Grigorchuk, Nekrashevych and Sushchansky in [11, Problem 7.2.1(a)] is that of *asynchronous automata*, namely automata given by $\Phi\colon A \times S \to S \times A^*$ that produce zero or more letters of output each time an input letter is read. For these automata, it was already shown by Belk and Bleak [3] that the order problem is undecidable.

Gillibert's proof of Theorem B uses a simulation of arbitrary Turing machines by transducers via cellular automata.

Acknowledgments. The authors are grateful to the anonymous referee for valuable references and remarks on Minsky machines that helped improve the clarity of the text.

This work is supported by the "@raction" grant ANR-14-ACHN-0018-01.

2. Functionally recursive groups and Minsky machines

All our theorems are proven by embedding Minsky machine computations into functionally recursive groups. Let us recall more precisely the definition of these machines:

Definition 2.1. A *Minsky machine* is a computational device **M** equipped with two integer counters m, n and a finite amount of additional memory. It has a finite set S of *states*, an *initial state* $s_* \in S$, a *final state* $s_\dagger \in S$, and for each state $s \neq s_\dagger$ an instruction, which can be any of the following kind:

I: $(s, m, n) \mapsto (s', m + 1, n)$;

II: $(s, m, n) \mapsto (s', m, n + 1)$;

III: $(s, m, n) \mapsto (s', m + 1, n + 1)$;

IV: $(s, m, n) \mapsto (s', m - 1, n)$, only valid if $m > 0$;

V: $(s, m, n) \mapsto (s', m, n - 1)$, only valid if $n > 0$;

VI: $(s, m, n) \mapsto (s', n, m)$;

VII: $(s, m, n) \mapsto (m = 0\,?\,s' : s'', m, n)$;

VIII: $(s, m, n) \mapsto (n = 0\,?\,s' : s'', m, n)$;

IX: $(s, m, n) \mapsto m = 0\,?\,(s', m, n) : (s'', m - 1, n)$;

X: $(s, m, n) \mapsto n = 0\,?\,(s', m, n) : (s'', m, n - 1)$.

(We use the C style "?:" operator, with '$a\,?\,b : c$' meaning 'if a then b else c'.)

As **M** is turned on, its state and counters initialize at $(s_0, m_0, n_0) = (s_*, 0, 0)$, and then $(s_{i+1}, m_{i+1}, n_{i+1})$ is determined from (s_i, m_i, n_i) using the prescribed rules. If at some moment $s_i = s_\dagger$ then **M** *stops*; otherwise it runs forever.

The machine **M** may also be treated as a machine with input, say $n \in \mathbb{N}$; it is then initialized at $(s_0, m_0, n_0) = (s_*, 0, n)$.

We recall the main result on Mealy machines, testifying to their computational power:

Proposition 2.2 ([18]). (1) *There is no algorithm that, given a Minksy machine* **M***, determines whether* **M** *stops.*

(2) *There is a "universal" Minsky machine* **M** *such that*

$$\{n \in \mathbb{N} \mid \mathbf{M} \textit{ stops when turned on in configuration } (s_*, 0, 2^n)\}$$

is not recursive.

Proposition 2.2 is proven (in [18, Theorem I]) by showing how an arbitrary Turing machine **T** may be emulated by a Minsky machine $\mathbf{M_T}$. Beware, however, that the input of **T** must be preprocessed before it is fed to $\mathbf{M_T}$; for instance, if **T** starts with $k \in \mathbb{N}$ on its tape, written in binary, then $\mathbf{M_T}$ should start with 2^k in its second counter.

Likewise, a universal Turing machine $\mathbf{T}_u$ (that receives on its input tape a description of a Turing machine **T** and simulates it) can be emulated by a universal Minsky machine $\mathbf{M}_u$, that starts with 2^k on its second counter, for k a Gödel-encoding of **T**.

Minsky machines with at least five counters are universal calculators; namely, for every primitive recursion function $\phi(n)$ there exists a five-counter Minsky machine that, when turned on in configuration $(s_*, 0, 0, 0, 0, n)$, stops in configuration $(s_\dagger, 0, 0, 0, 0, \phi(n))$. This is *not* true for two-counter Minsky machines: there does not, for example, exist a machine that starts in $(s_*, 0, n)$ and stops in $(s_\dagger, 0, 2^n)$, see [21]. However, a five-counter Minsky machine may be emulated by a two-counter Minsky machine, by representing its configuration (s, i, j, k, ℓ, m) as $(s, 0, 2^i 3^j 5^k 7^\ell 11^m)$ in the two-counter associated machine, see [19, Chapter 14] for details.

We finally note that only one of the instructions {I,II} and III is necessary, and that in the presence of VI only one of I,II, one of IV,V, one of VII,VIII and one of IX,X is necessary. Minimal sets of instructions are {III,IV,V,VII,VIII} and {I,IV,VI,VII} and {III,IX,X} and {I,VI,IX}.

2.1. Proof of Theorem A′. Let **M** be a Minsky machine with stateset $S_0 = \{s_i, s_j, \dots\}$. Without loss of generality, we assume that all instructions of **M** are of type I, VI, IX.

We construct a functionally recursive group $\langle \Phi_{\mathbf{M}} \rangle$ presented by $\Phi_{\mathbf{M}}\colon A \times S \to F_S \times A$, for sets A, S given as follows: the generating set S consists of

- elements x, y, $s_\dagger$, t and u;
- for each state $s_i \in S_0$ of type I or IX, an element s_i;
- for each state $s_i \in S_0$ of type VI, three elements s_i, a_i, b_i.

The alphabet A consists of

- four letters 0, 1, $\dagger_1$ and $\dagger_2$;
- for each state $s_i \in S_0$ of type I, a letter $\mathbf{i}_1$;
- for each state $s_i \in S_0$ of type IX, two letters $\mathbf{i}_1$ and $\mathbf{i}_2$;
- for each state $s_i \in S_0$ of type VI, five letters $\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_5$.

Our notation thus compactly associates a collection of alphabet letters with states of $\mathbf{M}$: its states $s_i, s_j, s_k, \dots$ correspond to alphabet letters $\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1, \dots, \mathbf{i}_2, \dots$; thus for instance with s_5 of type IX are associated letters $5_1, 5_2$.

The map $\Phi_{\mathbf{M}}\colon A \times S \to F_S \times A$ is given below, with ϵ denoting the empty word in F_S. Whenever a value of $\Phi_{\mathbf{M}}$ is unspecified, we take it to mean $\Phi_{\mathbf{M}}(a, s) = (s, a)$.

- For the states $s_\dagger$ and t, u we put

$$\begin{aligned}
&\Phi_{\mathbf{M}}(0, s_\dagger) = (\epsilon, \dagger_1); \quad \Phi_{\mathbf{M}}(\dagger_1, x) = (\epsilon, \dagger_1); \quad \Phi_{\mathbf{M}}(\dagger_1, y) = (\epsilon, \dagger_1);\\
&\Phi_{\mathbf{M}}(\dagger_1, s_\dagger) = (\epsilon, 0); \quad \Phi_{\mathbf{M}}(\dagger_2, x) = (\epsilon, \dagger_2); \quad \Phi_{\mathbf{M}}(\dagger_2, y) = (\epsilon, \dagger_2);\\
&\Phi_{\mathbf{M}}(\dagger_1, t) = (\epsilon, \dagger_2); \quad \Phi_{\mathbf{M}}(0, u) = (u, 1); \quad \Phi_{\mathbf{M}}(\dagger_2, s_\dagger) = (\epsilon, \dagger_2);\\
&\Phi_{\mathbf{M}}(\dagger_2, t) = (\epsilon, \dagger_1); \quad \Phi_{\mathbf{M}}(1, u) = (u, 0);
\end{aligned}$$

- for all $g \in S \setminus \{u\}$ we put $\Phi_{\mathbf{M}}(1, g) = (\epsilon, 1)$, and for all $a \in A \setminus \{0, 1\}$ we put $\Phi_{\mathbf{M}}(a, u) = (\epsilon, a)$;
- for each instruction $(s_i, m, n) \mapsto (s_j, m + 1, n)$ of type I we put

$$\begin{aligned}
&\Phi_{\mathbf{M}}(0, s_i) = (s_j, \mathbf{i}_1); \quad \Phi_{\mathbf{M}}(\mathbf{i}_1, s_i) = (\epsilon, 0);\\
&\Phi_{\mathbf{M}}(\mathbf{i}_1, x) = (x^2, \mathbf{i}_1); \quad \Phi_{\mathbf{M}}(\mathbf{i}_1, y) = (y, \mathbf{i}_1);
\end{aligned}$$

- for each instruction $(s_i, m, n) \mapsto (s_j, n, m)$ of type VI, $\Phi_{\mathbf{M}}(a, s)$ is written at position (a, s) of Table 1.

Table 1

element of S	input letter: 0	$\mathbf{i}_1$	$\mathbf{i}_2$	$\mathbf{i}_3$	$\mathbf{i}_4$	$\mathbf{i}_5$
x		$(x^{b_i x}, \mathbf{i}_1)$	$(\epsilon, \mathbf{i}_3)$	$(x, \mathbf{i}_2)$	$(x^2, \mathbf{i}_4)$	$(y, \mathbf{i}_5)$
y		$(y^x, \mathbf{i}_1)$	$(y, \mathbf{i}_2)$		$(y, \mathbf{i}_4)$	$(x, \mathbf{i}_5)$
s_i	$(a_i b_i x, \mathbf{i}_1)$	$(\epsilon, 0)$				
a_i	$(a_i, \mathbf{i}_2)$		$(\epsilon, 0)$			
b_i			$(b_i, \mathbf{i}_4)$	$(a_i^{-1} s_j, \mathbf{i}_5)$	$(\epsilon, \mathbf{i}_2)$	$(\epsilon, \mathbf{i}_3)$

- for each instruction $(s_i, m, n) \mapsto (m = 0\,?\,s_j : s_k, \max(0, m-1), n)$ of type IX we put

$$\Phi_{\mathbf{M}}(0, s_i) = (s_k, \mathbf{i}_1); \quad \Phi_{\mathbf{M}}(\mathbf{i}_1, x) = (s_k^{-1} s_j x, \mathbf{i}_2); \quad \Phi_{\mathbf{M}}(\mathbf{i}_1, y) = (y, \mathbf{i}_1);$$
$$\Phi_{\mathbf{M}}(\mathbf{i}_1, s_i) = (\epsilon, 0); \quad \Phi_{\mathbf{M}}(\mathbf{i}_2, x) = (x^{-1} s_j^{-1} s_k x, \mathbf{i}_1); \quad \Phi_{\mathbf{M}}(\mathbf{i}_2, y) = (y, \mathbf{i}_2).$$

Theorem A′ follows from the undecidability of the halting problem for the Minsky machines (Proposition 2.2) and the following

Proposition 2.3. *Consider the infinite sequence $W = 0^\infty$. Then the Minsky machine* **M** *does not halt if and only if the action of $\langle \Phi_{\mathbf{M}} \rangle$ satisfies*

$$W^{(s_* x y) t (s_* x y)^{-1}} = W.$$

Proof. We encode the states of **M** by elements of F_S. The word

$$(s_i x^{2^m} y^{2^n}) t (s_i x^{2^m} y^{2^n})^{-1}$$

corresponds to the configuration (s_i, m, n).

It is convenient to write $\Phi_{\mathbf{M}}(a, g) = (g', a')$ in the form $a \cdot g = g' \cdot a'$. In this manner, the computation of the functionally recursive action is given by a sequence of exchanges of letters with words in F_S. We check the following equalities:

If $(s_i, m, n) \to (s_j, m+1, n)$ is an instruction of type I, then

$$0 \cdot (s_i x^{2^m} y^{2^n}) t (s_i x^{2^m} y^{2^n})^{-1} = (s_j x^{2^{m+1}} y^{2^n}) t (s_j x^{2^{m+1}} y^{2^n})^{-1} \cdot 0. \tag{2.1}$$

Indeed $0 \cdot s_i x^{2^m} y^{2^n} = s_j \cdot \mathbf{i}_1 \cdot x^{2^m} y^{2^n} = s_j x^{2^{m+1}} y^{2^n} \cdot \mathbf{i}_1$; the claim follows from $\mathbf{i}_1 \cdot t = t \cdot \mathbf{i}_1$ and the reverse $\mathbf{i}_1 \cdot (s_i x^{2^m} y^{2^n})^{-1} = (s_j x^{2^{m+1}} y^{2^n})^{-1} \cdot 0$.

If $(s_i, m, n) \to (s_j, n, m)$ is an instruction of type VI, then

$$0^{m+2} \cdot (s_i x^{2^m} y^{2^n}) t (s_i x^{2^m} y^{2^n})^{-1} = (s_j x^{2^n} y^{2^m}) t (s_j x^{2^n} y^{2^m})^{-1} \cdot 0^{m+2}. \tag{2.2}$$

Indeed we first check $0 \cdot s_i x^{2^m} y^{2^n} = a_i b_i x (x^{b_i x})^{2^m} (y^x)^{2^n} \cdot \mathbf{i}_1 = a_i x^{2^m} b_i y^{2^n} x \cdot \mathbf{i}_1$.

We obtained a word with two "blocks" of x: the blocks x^{2^m} and x^{2^0}. Each time a '0' letter is multiplied on the left of that word, the size of the first block will halve and the size of the second one will double: for $m, n, p \in \mathbb{N}$, we have

$$0 \cdot a_i x^{2m} b_i y^n x^p = a_i x^m b_i y^n x^{2p} \cdot \mathbf{i}_4$$

so $0^{m+1} \cdot s_i x^{2^m} y^{2^n} = a_i x b_i y^{2^n} x^{2^m} \cdot (\mathbf{i}_4)^m \mathbf{i}_1$. Then

$$0 \cdot a_i x b_i y^{2^n} x^{2^m} = a_i (a_i^{-1} s_j) x^{2^n} y^{2^m} \cdot \mathbf{i}_5,$$

so $0^{m+2}\cdot s_i x^{2^m} y^{2^n} = s_j x^{2^n} y^{2^m}\cdot \mathbf{i}_5(\mathbf{i}_4)^m \mathbf{i}_1$. Recalling that we have $a\cdot t = t\cdot a$ for all $a = \mathbf{i}_1,\dots,\mathbf{i}_5$, the claim is proven.

If $(s,m,n)\to(m=0\,?\,s_j : s_k, \max(m-1,0), n)$ is an instruction of type IX, then if $m=0$ we have

$$0\cdot(s_i x^{2^m} y^{2^n}) t (s_i x^{2^m} y^{2^n})^{-1} = (s_j x^{2^m} y^{2^n}) t (s_j x^{2^m} y^{2^n})^{-1}\cdot 0 \tag{2.3}$$

while if $m>0$ we have

$$0\cdot(s_i x^{2^m} y^{2^n}) t (s_i x^{2^m} y^{2^n})^{-1} = (s_k x^{2^{m-1}} y^{2^n}) t (s_k x^{2^{m-1}} y^{2^n})^{-1}\cdot 0. \tag{2.4}$$

Indeed in the first case we have

$$0\cdot s_i x y^{2^n} = s_k (s_k^{-1} s_j x) y^{2^n}\cdot \mathbf{i}_2,$$

while in the second case we have

$$0\cdot s_i x^{2^m} y^{2^n} = s_k (s_j^{-1} s_k x\cdot x^{-1} s_k^{-1} s_j x)^{2^{m-1}} y^{2^n}\cdot \mathbf{i}_1 = s_k x^{2^{m-1}} y^{2^n}\cdot \mathbf{i}_1.$$

Recalling that we have $a\cdot t = t\cdot a$ for all $a=\mathbf{i}_1,\mathbf{i}_2$, the claim is proven.

From (2.1)–(2.4) it follows that if $\mathbf{M}$ does not halt then $W^{(s_* xy) t (s_* xy)^{-1}} = W$. Conversely, if $\mathbf{M}$ halts then there exist $k,m,n\in\mathbb{N}$ such that

$$0^k\cdot(s_* xy) t (s_* xy)^{-1} = (s_\dagger x^{2^m} y^{2^n}) t (s_\dagger x^{2^m} y^{2^n})^{-1}\cdot 0^k.$$

Then

$$0\cdot s_\dagger x^{2^m} y^{2^n} t (s_\dagger x^{2^m} y^{2^n})^{-1} = \dagger_1\cdot t (s_\dagger x^{2^m} y^{2^n})^{-1} = \dagger_2\cdot (s_\dagger x^{2^m} y^{2^n})^{-1} = \dagger_2.$$

In that case, we have $W^{(s_* xy) t (s_* xy)^{-1}} = 0^k \dagger_2 0^\infty \neq W$. □

The computations are best carried on $\Phi_{\mathbf{M}}$'s *dual Moore diagram* Δ, see Figure 2: this is the directed labeled graph with vertex set A and with for all $a\in A, s\in S$ an edge from a to b labeled (s,t) whenever $\Phi_{\mathbf{M}}(a,s) = (t,b)$. One checks an equality '$\Phi_{\mathbf{M}}(a,s)=(t,b)$' by finding in Δ a path starting at a with input label s; the endpoint of the path is b, and the output label is t.

2.2. Proof of Theorem A. We have not yet used the letter 1 and the state u of $\Phi_{\mathbf{M}}$. Theorem A follows now from the following

Proposition 2.4. *The Minsky machine* $\mathbf{M}$ *halts if and only if*

$$[(s_* xy) t (s_* xy)^{-1}, u] \neq 1 \text{ in } \langle \Phi_{\mathbf{M}}\rangle.$$

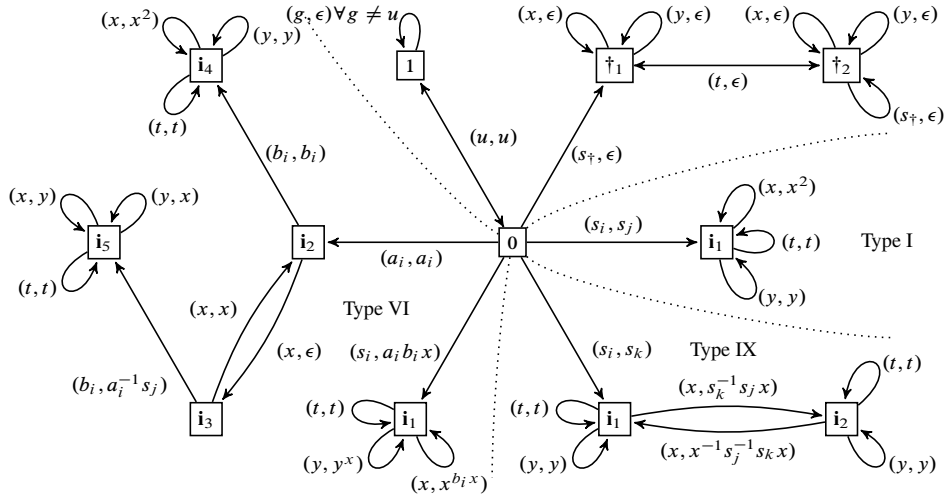
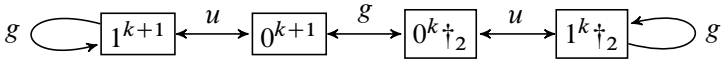


Figure 2. The dual Moore diagram of $\Phi_{\mathbf{M}}$, used in the proof of Theorem A

Proof. The element u acts on A^ω as follows: it scans $X \in A^\omega$ for its longest prefix in $\{0, 1\}^*$, and exchanges all 0's and 1's in that prefix. Write $g = (s_* xy)t(s_* xy)^{-1}$; from Proposition 2.3 we know that g fixes 0^∞ if and only if $\mathbf{M}$ does not halt.

Assume first that $\mathbf{M}$ does not halt; then g in fact also fixes $\{0, 1\}^\infty$, so the supports of g and u are disjoint and $[g, u] = 1$ in $\langle \Phi_{\mathbf{M}} \rangle$.

Assume next that $\mathbf{M}$ does halt; without loss of generality, we may assume $\mathbf{M}$ does not stop immediately, so there is $k \geq 1$ such that $(0^{k+1})^g = 0^k \dagger_2$. Since $(0^{k+1})^u = 1^{k+1}$ and $(0^k \dagger_2)^u = 1^k \dagger_2$ and $(1^{k+1})^g = 1^{k+1}$ and $(1^k \dagger_2)^g = 1^k \dagger_2$, the commutator $[g, u]$ acts as a 2-2-cycle $(0^{k+1}, 0^k \dagger_2)(1^{k+1}, 1^k \dagger_2)$ and in particular $[g, u] \neq 1$ in $\langle \Phi_{\mathbf{M}} \rangle$:



□

2.3. Proof of Theorem A″. Consider a Minsky machine $\mathbf{M}_u$ such that $\{n \in \mathbb{N} \mid \mathbf{M}_u$ halts when started in configuration $(s_*, 0, 2^n)\}$ is not recursive, see Proposition 2.2(2). Theorem A″ follows by considering in the group $\langle \Phi_{\mathbf{M}_u} \rangle$ the elements $[(s_* xy^{2^n})t(s_* xy^{2^n})^{-1}, u]$; this set of words is recursive, but the subset of those that equal 1 in $\langle \Phi_{\mathbf{M}_u} \rangle$ is not recursive.

3. Automata groups and Minsky machines

3.1. Proof of Theorem B. Let $\mathbf{M}$ by a Minsky machine with stateset S_0. Without loss of generality, we assume that all instructions of $\mathbf{M}$ are of type III,IV,V,VII,VIII, as defined in the beginning of Section 2, so

$$S_0 = S_{\mathrm{III}} \sqcup S_{\mathrm{IV}} \sqcup S_{\mathrm{V}} \sqcup S_{\mathrm{VII}} \sqcup S_{\mathrm{VIII}} \sqcup \{s_\dagger\}.$$

We consider the transducer with stateset $S := S_0^{\pm 1} \sqcup \{\epsilon, x, x^{-1}, y, y^{-1}\}$ and alphabet

$$A = \{\mathrm{III}_i, \mathrm{IV}_i, \mathrm{V}_i, \mathrm{VII}_j, \mathrm{VIII}_j \mid i = 1, 2, \bar{1}, \bar{2};\ j = 1, \dots, 4, \bar{1}, \dots, \bar{4}\}.$$

The structure of the transducer is given by its map $\Phi_{\mathbf{M}}\colon A \times S \to S \times A$, first described as a table, with $\Phi_{\mathbf{M}}(a, s)$ at position (a, s). The state ϵ is the identity, and $\Phi_{\mathbf{M}}(a, \epsilon) = (\epsilon, a)$ for all $a \in A$.

For all instructions $(s, m, n) \mapsto (s', m + 1, n + 1)$ of type III and for all $t \in S_0 \setminus S_{\mathrm{III}}$ we have Table 2.

Table 2

		input letter			
		III_1	III_2	$\mathrm{III}_{\bar{1}}$	$\mathrm{III}_{\bar{2}}$
in state	x	(x, III_1)	(x, III_2)	$(x^{-1}, \mathrm{III}_{\bar{1}})$	$(x^{-1}, \mathrm{III}_{\bar{2}})$
	y	(y, III_1)	(y, III_2)	$(y^{-1}, \mathrm{III}_{\bar{1}})$	$(y^{-1}, \mathrm{III}_{\bar{1}})$
	s	(s', III_2)	$(\epsilon, \mathrm{III}_1)$	$(\epsilon, \mathrm{III}_{\bar{2}})$	$((s')^{-1}, \mathrm{III}_{\bar{1}})$
	t	$(\epsilon, \mathrm{III}_{\bar{1}})$	$(\epsilon, \mathrm{III}_{\bar{2}})$	$(\epsilon, \mathrm{III}_1)$	$(\epsilon, \mathrm{III}_2)$

For all instructions $(s, m, n) \mapsto (s', m-1, n)$ of type IV and for all $t \in S_0 \setminus S_{\mathrm{IV}}$, we have Table 3. The same applies for an instruction of type V, with the roles of x, y switched.

Table 3

		input letter			
		IV_1	IV_2	$\mathrm{IV}_{\bar{1}}$	$\mathrm{IV}_{\bar{2}}$
in state	x	(x, IV_2)	$(\epsilon, \mathrm{IV}_1)$	$(\epsilon, \mathrm{IV}_{\bar{2}})$	$(x^{-1}, \mathrm{IV}_{\bar{1}})$
	y	(y, IV_1)	(y, IV_2)	$(y^{-1}, \mathrm{IV}_{\bar{1}})$	$(y^{-1}, \mathrm{IV}_{\bar{2}})$
	s	(s', IV_1)	(s', IV_2)	$((s')^{-1}, \mathrm{IV}_{\bar{1}})$	$((s')^{-1}, \mathrm{IV}_{\bar{2}})$
	t	$(\epsilon, \mathrm{IV}_{\bar{1}})$	$(\epsilon, \mathrm{IV}_{\bar{2}})$	$(\epsilon, \mathrm{IV}_1)$	$(\epsilon, \mathrm{IV}_2)$

For an instruction $(s, m, n) \mapsto (m = 0\,?\,s' : s'', m, n)$ of type VII and for all $t \in S_0 \setminus S_{\mathrm{VII}}$, we have Table 4. The same applies for an instruction of type VIII, with the roles of x, y switched. Note that $s_\dagger$ is treated as a state t in all tables above.

Table 4

		input letter							
		VII_1	VII_2	VII_3	VII_4	$\mathrm{VII}_{\bar{1}}$	$\mathrm{VII}_{\bar{2}}$	$\mathrm{VII}_{\bar{3}}$	$\mathrm{VII}_{\bar{4}}$
in state	x	(x, VII_4)	$(\epsilon, \mathrm{VII}_3)$	$(\epsilon, \mathrm{VII}_2)$	(x, VII_1)	$(x^{-1}, \mathrm{VII}_{\bar{4}})$	$(\epsilon, \mathrm{VII}_{\bar{3}})$	$(\epsilon, \mathrm{VII}_{\bar{2}})$	$(x^{-1}, \mathrm{VII}_{\bar{1}})$
	y	(y, VII_1)	$(\epsilon, \mathrm{VII}_2)$	$(\epsilon, \mathrm{VII}_3)$	(y, VII_4)	$(y^{-1}, \mathrm{VII}_{\bar{1}})$	$(\epsilon, \mathrm{VII}_{\bar{2}})$	$(\epsilon, \mathrm{VII}_{\bar{3}})$	$(y^{-1}, \mathrm{VII}_{\bar{4}})$
	s	$(\epsilon, \mathrm{VII}_2)$	(s'', VII_1)	(s', VII_4)	$(\epsilon, \mathrm{VII}_{\bar{4}})$	$((s'')^{-1}, \mathrm{VII}_{\bar{2}})$	$(\epsilon, \mathrm{VII}_{\bar{1}})$	$(\epsilon, \mathrm{VII}_3)$	$((s')^{-1}, \mathrm{VII}_{\bar{3}})$
	t	$(\epsilon, \mathrm{VII}_{\bar{1}})$	$(\epsilon, \mathrm{VII}_{\bar{2}})$	$(\epsilon, \mathrm{VII}_{\bar{3}})$	$(\epsilon, \mathrm{VII}_{\bar{4}})$	$(\epsilon, \mathrm{VII}_1)$	$(\epsilon, \mathrm{VII}_2)$	$(\epsilon, \mathrm{VII}_3)$	$(\epsilon, \mathrm{VII}_4)$

Theorem B follows from the undecidability of the halting problem for Minsky machines, and the following

Proposition 3.1. *The Minsky machine* $\mathbf{M}$ *constructed above halts if and only if the element* s_*xy *has finite order in* $\langle\Phi_{\mathbf{M}}\rangle$.

Proof. Set $G = \langle\Phi_{\mathbf{M}}\rangle$. For $g \in G$, denote by $C(g)$ its symmetrized conjugacy class:[1]

$$C(g) := \{g^{\pm x} \mid x \in G\}.$$

Given a symmetrized conjugacy class C, choose a representative g in it, let $A = A_1 \sqcup \cdots \sqcup A_\ell$ be the decomposition of A into cycles for the action of g, and choose representatives $a_i \in A_i$. We have $\bar{\Phi}_{\mathbf{M}}(a_i, g^{\#A_i}) = (h_i, a_i)$ for some $h_i \in G$, and it is easy to see that the collection of symmetrized conjugacy class $\{C(h_i) \mid i = 1, \dots, \ell\}$ is independent of the choice of g and the a_i.

We construct an integer-labeled, directed graph[2] whose vertices are symmetrized conjugacy classes in G; for a conjugacy class C as above, there are ℓ edges starting at C, ending respectively at $C(h_1), \dots, C(h_\ell)$ with labels $\#A_1, \dots, \#A_\ell$.

Lemma 3.2. *For* $g \in G$, *its order* (*in* $\mathbb{N} \cup \{\infty\}$) *is the least common multiple, along all paths starting at* $C(g)$, *of the product of the labels along the path.*

Proof. Consider a path starting at $C(g)$, with labels $n_1, \dots, n_s$, and going through vertices $C(g_1), \dots, C(g_s)$. Then g has an orbit of length n_1 on A, so the order of g is a multiple of n_1. Furthermore, g^{n_1} fixes pointwise that orbit, and acts as an element of $C(g_1)$ on any sequence that starts by a letter in that orbit. Recursively, the order of g_1 is a multiple of $n_2 \cdots n_s$, so the order of g is a multiple of $n_1 \cdots n_s$. In particular, if there are paths with arbitrarily large product of labels then g has infinite order.

Conversely, if g has infinite order then there are arbitrarily long orbits of g on A^*, so there are paths with arbitrarily large product of labels; and if m be the least common multiple of all path labels then all edges on paths starting at $C(g^m)$ are labeled 1 so g^m fixes every sequence and therefore $g^m = 1$. △

Let us compute the subgraph spanned by $C(s_*xy)$. For the computations, it is helpful to picture the operation of the transducer $\Phi_{\mathbf{M}}$ by means of its dual Moore diagram Δ, see Figure 3. Given $g \in G$, we compute all primitive cycles in Δ

[1] The reason we consider symmetrized conjugacy classes is that every element of $C(g)$ has same order, and a process will naturally produce symmetrized conjugacy classes out of symmetrized conjugacy classes, but would not be well-defined at the level of usual conjugacy classes.

[2] This graph essentially appears in the solution of [5] to the order problem in bounded automata.

whose input label is a power of g, and read the corresponding output label; these are the h_i in the map on symmetrized conjugacy classes $C(g) \rightsquigarrow \{C(h_i)\}$.

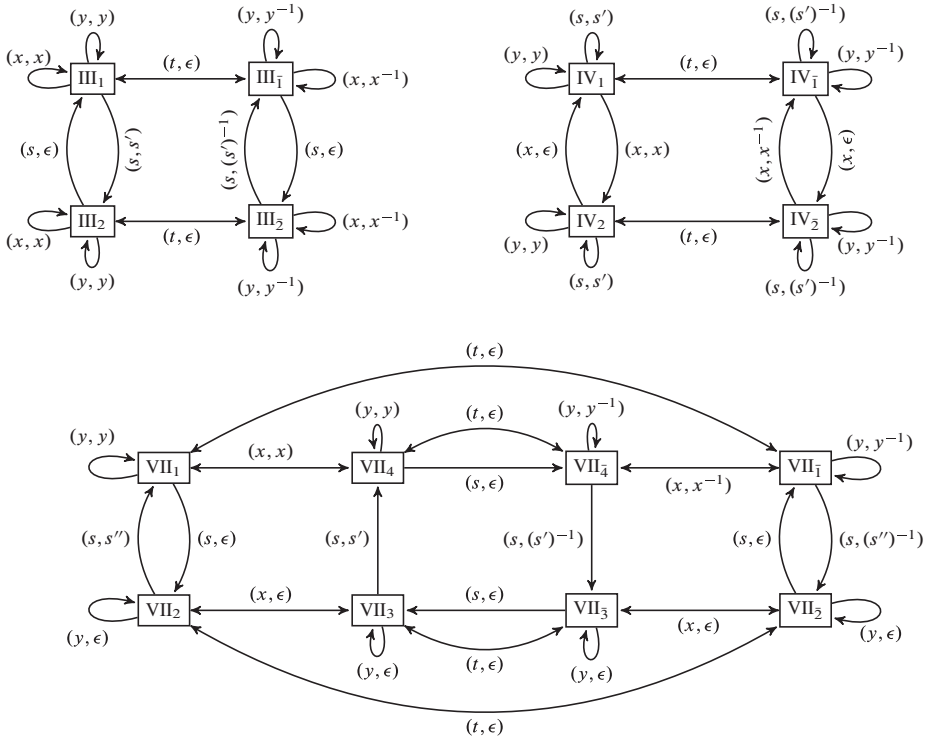


Figure 3. The dual Moore diagram of $\Phi_{\mathbf{M}}$, used in the proof of Theorem B

We first note, by direct inspection, that x and y commute in G. This follows by tracing the path $x^{-1}y^{-1}xy$ in the graphs above, and noting that they always induce the trivial permutation of A with output either trivial or conjugate to $(x^{-1}y^{-1}xy)^{\pm 1}$.

We now claim that, if $(s, m, n) \to (s', m', n')$ is a transition of the machine $\mathbf{M}$, then the conjugacy class $C(sx^{2^m}y^{2^n})$ has at least one arrow to $C(s'x^{2^{m'}}y^{2^{n'}})$, and possibly other arrows, all of them to $C(1)$. We also claim that if s is not of type IV or V, then arrows to $C(s'x^{2^{m'}}y^{2^{n'}})$ are with labels > 1; and all arrows from $C(s_\dagger x^{2^m}y^{2^n})$ are arrows to $C(1)$. We see that if the machine halts then every path starting at $C(s_* xy)$ has only a finite number of labels > 1, and this shows that the order of $s_* xy$ is finite.

On the other hand, if the machine does not halt then there is an path with infinitely many labels > 1 (because no Minsky machine can decrease its counters infinitely many times in a row) so $s_* xy$ has infinite order.

Note that our transducer has the property $\Phi_{\mathbf{M}}(L_{\bar{i}}, g) = \Phi_{\mathbf{M}}(L_i, g^{-1})$, for all $g \in S$ and all $L \in \{\mathrm{III}, \mathrm{IV}, \mathrm{V}, \mathrm{VII}, \mathrm{VIII}\}$. Also note that $\Phi_{\mathbf{M}}(L_i, t) = \Phi_{\mathbf{M}}(L_i, t^{-1}) = (\epsilon, L_{\bar{i}})$ whenever t is any instruction not of type L.

Using this, we can prove that if t is not of type L, then $tg_ng_{n-1}\cdots g_1tg_1g_2\cdots g_n$ fixes the orbit $\{L_i\}$ with output ϵ. Indeed,

$$tg_ng_{n-1}\cdots g_1tg_1g_2\cdots g_n = (g_ntg_nt^{-1}\cdot tg_{n-1}g_{n-2}\cdots g_1tg_1g_2\cdots g_{n-1})^{g_n},$$

and we use induction on n. It follows that $(tx^my^n)^2$ fixes L_i with outputs ϵ, i.e., there is an arrow from $C(tx^my^n)$ to $C(1)$ with label 2.

Let us first restrict to the orbit $\{\mathrm{III}_i\}$ of G on A. We consider $g := sx^my^n$ with s an instruction of type III. It acts as a product of two cycles $(\mathrm{III}_1, \mathrm{III}_2)(\mathrm{III}_{\bar{1}}, \mathrm{III}_{\bar{2}})$; the output label of g^2 on the first cycle, starting at III_1, is

$$s'x^my^n\epsilon x^my^n = s'x^{2m}y^{2n},$$

and the output of g^2 starting on the second cycle at $\mathrm{III}_{\bar{1}}$ is

$$\epsilon x^{-m}y^{-n}(s')^{-1}x^{-m}y^{-n} \in C(s'x^{2m}y^{2n}).$$

There are therefore two arrows from $C(sx^{2^m}y^{2^n})$ to $C(s'x^{2^{m+1}}y^{2^{n+1}})$, as required. We do not consider $g := tx^my^n$ with t an instruction of different type or $s_\dagger$, because it was considered above (there are some arrows to $C(1)$ with labels 2).

We restrict next to the orbit $\{\mathrm{IV}_i\}$ of G and consider the case $g = sx^{2m}y^n$. (We do not need to consider cases $g = sx^{2m+1}y^n$ or $g = tx^my^n$, for the first because we suppose that if $m = 0$ then $\mathbf{M}$ does not perform an instruction of type IV, and for the second because it was already considered above.)

An element $g = sx^{2m}y^n$ fixes IV_1, IV_2, $\mathrm{IV}_{\bar{1}}$ and $\mathrm{IV}_{\bar{2}}$, its outputs are respectively $s'(x\epsilon)^my^n = s'x^my^n$, $s'x^my^n$, $(s')^{-1}x^{-m}y^{-n}$ and $(s')^{-1}x^{-m}y^{-n}$. Hence there are four arrows from $C(sx^{2^m}y^{2^n})$ to $C(s'x^{2^{m-1}}y^{2^n})$, all with labels 1.

We restrict next to the orbit $\{\mathrm{VII}_i\}$ of G and perform the same computations; the result is in Table 5.

Table 5

$g \in G$	cycles of g	output, starting at first element of the cycle
$sx^{2m}y^n$	$(\mathrm{VII}_1, \mathrm{VII}_2)$	$\epsilon s''x^{2m}y^n$
	$(\mathrm{VII}_3, \mathrm{VII}_4, \mathrm{VII}_{\bar{4}}, \mathrm{VII}_{\bar{3}})$	$s'x^{2m}y^n\epsilon x^{-2m}y^{-n}(s')^{-1}\epsilon = 1$
	$(\mathrm{VII}_{\bar{1}}, \mathrm{VII}_{\bar{2}})$	$(s'')^{-1}\epsilon x^{-2m}y^{-n}$
$sx^{2m+1}y^n$	$(\mathrm{VII}_1, \mathrm{VII}_3)$	$\epsilon s'x^{2m+1}y^n$
	$(\mathrm{VII}_2, \mathrm{VII}_4, \mathrm{VII}_{\bar{1}}, \mathrm{VII}_{\bar{3}})$	$s''x^{2m+1}y^n\epsilon x^{-2m-1}y^{-n}(s'')^{-1}\epsilon = 1$
	$(\mathrm{VII}_{\bar{2}}, \mathrm{VII}_{\bar{4}})$	$\epsilon x^{-2m-1}y^{-n}(s')^{-1}\epsilon$

If $m > 0$ then there are two arrows from $C(sx^{2^m}y^{2^n})$ to $C(s''x^{2^m}y^{2^n})$ with label 2 and an arrow to $C(1)$ with label 4; if $m = 0$ then there are two arrows from $C(sx^{2^m}y^{2^n})$ to $C(s''x^{2^m}y^{2^n})$ with label 4 and an arrow to $C(1)$ with label 4.

The orbits $\{\mathrm{V}_i\}$ and $\{\mathrm{VIII}_i\}$ are investigated in the same way as $\{\mathrm{IV}_i\}$ and $\{\mathrm{VII}_i\}$ respectively. □

3.2. Proof of Theorem B″. Consider a Minsky machine $\mathbf{M}_u$ such that $\{n \in \mathbb{N} \mid \mathbf{M}_u \text{ halts when started in configuration } (s_*, 0, 2^n)\}$ is not recursive, see Proposition 2.2(2). Theorem B″ follows by considering in the group $\langle \Phi_{\mathbf{M}_u} \rangle$ the elements $s = s_*x$ and $t = y$.

3.3. Proof of Theorem B′. Let $\mathbf{M}$ be a Minsky machine with stateset S_0. Without loss of generality, we assume that all instructions are of type III,IX,X.

We associate to it the transducer with stateset

$$Q := S_0 \sqcup \{\epsilon, x, y\}$$

and alphabet

$$A = \{0, \mathrm{III}_i, \mathrm{IX}_j, \mathrm{X}_j \mid i = 1, 2;\ j = 1, \ldots, 4\}.$$

The structure of the transducer is given by its map $\Phi_{\mathbf{M}}\colon A \times Q \to Q \times A$. The state ϵ is the identity, and $\Phi_{\mathbf{M}}(a, \epsilon) = (\epsilon, a)$ for all $a \in A$.

- For all instructions

$$(s_i, m, n) \longmapsto (s_i', m + 1, n + 1)$$

of type III we have Table 6 and every instruction t of another type acts as $\Phi_{\mathbf{M}}(t, \mathrm{III}_\ell) = (\mathrm{III}_\ell, t)$.

Table 6

		input letter		
		0	III_1	III_2
in state	x	$(\epsilon, 0)$	(x, III_1)	(x, III_2)
	y	$(\epsilon, 0)$	(y, III_1)	(y, III_2)
	s_i	(s_i', III_1)	$(\epsilon, \mathrm{III}_2)$	$(\epsilon, 0)$

- For all instructions

$$(s_j, m, n) \longmapsto m = 0\,?\,(s_j', m, n) : (s_j'', m - 1, n)$$

of type IX we have Table 7 and every instruction t of another type acts as $\Phi_{\mathbf{M}}(\mathrm{IX}_\ell, t) = (t, \mathrm{IX}_\ell)$.

Table 7

		input letter				
		0	IX_1	IX_2	IX_3	IX_4
in state	x	$(\epsilon, 0)$	$(\epsilon, \mathrm{IX}_2)$	$(\epsilon, \mathrm{IX}_1)$	(x, IX_4)	$(\epsilon, \mathrm{IX}_3)$
	y	$(\epsilon, 0)$	$(\epsilon, \mathrm{IX}_1)$	$(\epsilon, \mathrm{IX}_2)$	(y, IX_3)	(y, IX_4)
	s_j	$(\epsilon, \mathrm{IX}_1)$	(s_j'', IX_4)	(s_j', IX_3)	(s_j', IX_2)	$(\epsilon, 0)$

- The same applies for every instruction

$$(s_k, m, n) \longmapsto n = 0\,?\,(s_k', m, n) : (s_k'', m, n-1)$$

  of type X, with the roles of x and y switched.

- For all $a \in A$ we have $\Phi_{\mathbf{M}}(a, s_\dagger) = (\epsilon, a)$.

We claim that the orbit of 0^∞ under $s_* xy$ is finite if and only if the machine $\mathbf{M}$ stops.

Set $G = \langle \Phi_{\mathbf{M}} \rangle$. We construct an integer-labeled, directed graph whose vertices are elements of G. For $g \in G$ consider its action on A and the minimal p_g such that g^{p_g} fixes 0, i.e., $0 \cdot g^{p_g} = g' \cdot 0$. In our graph we put an edge $g \to g'$ with label p_g on it.

The size of the the orbit of 0^∞ under $s_* xy$ is a finite number or ∞ and it is equal to the product of the labels along the path starting at $s_* xy$.

We claim that for any instruction $(s, m, n) \to (s', m', n')$ there is an edge from $sx^{2^m} y^{2^n}$ to $s' x^{2^{m'}} y^{2^{n'}}$ with label 3, and an edge from $s_\dagger x^{2^m} y^{2^n}$ to 1. This is checked on the dual Moore diagram of $\Phi_{\mathbf{M}}$, see Figure 4:

We first note that x and y commute in G. If $g = s_i x^m y^n$ and s_i is an instruction of type III, then the orbit of 0 under the action of g is $(0, \mathrm{III}_1, \mathrm{III}_2)$. There is an edge labeled 3 from g to $s_i' x^m y^n \epsilon x^m y^n = s_i' x^{2m} y^{2n}$.

Consider next s_j an instruction of type IX. There are two cases: if $g = s_j xy^n$ then the orbit of 0 is $(0, \mathrm{IX}_2, \mathrm{IX}_4)$ and the output is $\epsilon s_j' xy^n \epsilon$; if $g = s_j x^{2m} y^n$ then the orbit of 0 is $(0, \mathrm{IX}_1, \mathrm{IX}_4)$ and the output is $\epsilon s_j'' x^m y^n \epsilon$.

This means that if $m = 0$ then there is an edge labeled 3 from $s_j x^{2^m} y^{2^n}$ to $s_j' x^{2^m} y^{2^n}$, and if $m > 0$ then there is an edge labeled 3 from $s_j x^{2^m} y^{2^n}$ to $s_j'' x^{2^{m-1}} y^{2^n}$.

The same naturally applies to instructions of type IX. Finally, for all $m, n \in \mathbb{N}$ the element $s_\dagger x^m y^n$ fixes 0, and there is an edge labeled 1 from $s_\dagger x^m y^n$ to 1.

3.4. Contracting automata: proof of Theorem C. We finally explain how to make the transducers $\Phi_{\mathbf{M}}$ of the previous subsections contracting. We expand the definition from the introduction:

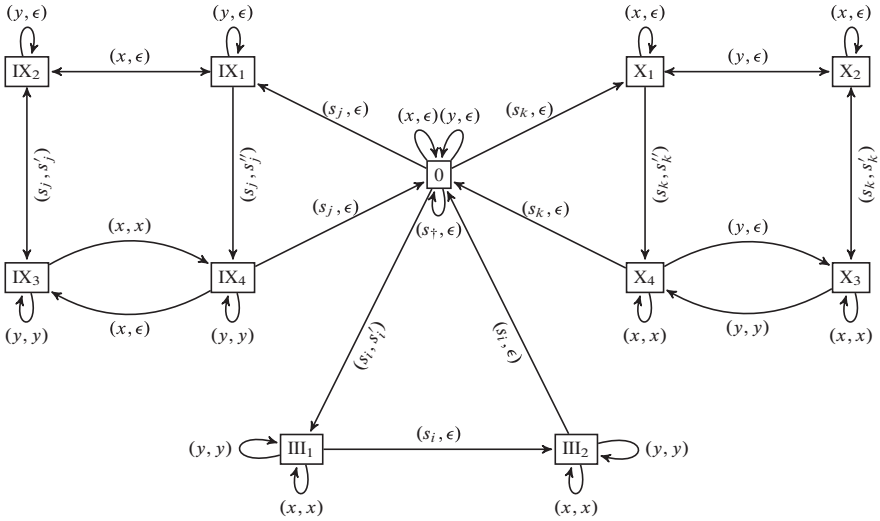


Figure 4. The dual Moore diagram of $\Phi_{\mathbf{M}}$, used in the proof of Theorem B′

Definition 3.3 ([20, Definition 2.11.1]). Let $G = \langle \Phi \rangle$ be a self-similar automata group with $\Phi\colon A \times S \to S \times A$ and $\bar{\Phi}\colon A \times G \to G \times A$. For $g \in G$ and $u \in A^*$, the *state* $g@u$ is the unique element of G such that $(uv)^g = u^g v^{g@u}$; namely, the action of g on the tails of sequences starting with u.

The group G is *contracting* if there exists a finite subset $N \subseteq G$ such that, for all $g \in G$, there exists $n(g) \in \mathbb{N}$ such that $g@u \in N$ whenever $|u| \geq n(g)$.

The minimal subset N satisfying the definition is called the *nucleus*. In particular, one has $n@a \in N$ for all $(a, n) \in A \times N$, so Φ induces an automaton still written $\Phi\colon A \times N \to N \times A$. Up to replacing S by $\widetilde{S} := S \cup N$ and A by $\widetilde{A} := A^n$ for n larger than $\max_{g \in \widetilde{S}^2} n(g)$, thus making the transducer process n letters at a time, one may also assume

$$\bar{\Phi}(\widetilde{A} \times \widetilde{S}^2) \subseteq \widetilde{S} \times \widetilde{A}.$$

A transducer Φ with this extra property is called *nuclear*.

Note that is is probably undecidable whether a self-similar group $\langle \Phi \rangle$ is contracting; but it is easy to decide whether a transducer $\Phi\colon A \times S \to S \times A$ is nuclear: by minimizing the composite transducer $A \times S^3 \to S \times A \times S^2 \to S^2 \times A \times S \to S^3 \times A$, find the set $\mathcal{R}$ of all words $s_1 s_2 s_3 \in S^3$ that equal 1 in G. Then Φ is nuclear if and only if for all $a \in A, s_1, s_2 \in S$ there exists $s_3 \in S$ such that if $\Phi(a, s_1) = (s'_1, b)$ and $\Phi(b, s_2) = (s'_2, c)$ then $s'_1 s'_2 s_3^{-1} \in \mathcal{R}$. The more precise form of Theorem C is:

Theorem 3.4. *There is no algorithm that, given a nuclear transducer $\Phi\colon A \times S \to S \times A$ and $a \in A$ and $s \in S$, determines the cardinality of the orbit of a^∞ under $\langle s \rangle$.*

There is no algorithm that, given a nuclear transducer $\Phi\colon A\times S\to S\times A$ *and* $s\in S$*, determines the order of* s *in* $\langle\Phi\rangle$.

Note that the group is not changed by these operations of replacing S by N and A by A^n. If Φ is nuclear, then $\langle\Phi\rangle$ is contracting in the sense of the introduction, since $|g'|\le(|g|+1)/2$ in the word metric defined by N. Conversely, if $|g'|\le\lambda|g|+C$ then one may take $N=\{g\in G\mid C/(1-\lambda)\ge|g|\}$ to see that G is contracting in the sense of Definition 3.3.

Lemma 3.5. *Let* $\Phi\colon A\times S\to S\times A$ *be a transducer. If there is a constant* $n\in\mathbb{N}$ *such that every reduced path of length* $\ge n$ *in the dual Moore diagram of* Φ *contains an* ϵ *letter along its output, then* $\langle\Phi\rangle$ *is contracting.*

Proof. Consider $g\in\langle\Phi\rangle$, and represent it by a word $w\in S^*$ of length $\ell=|g|$. Factor $w=w_1\dots w_t$ with $|w_i|=n$ for all $i=1,\dots,t-1$ and $|w_t|<n$.

Then every g' as in (1.1) is computed by following, in the dual Moore diagram, the path starting at a_1 with label w on its input. The output label along that path is g', and by hypothesis each time a segment w_i is read, for $i<t$, an ϵ letter is produced for g'; so $|g'|\le\ell-t+1$. Now $t=\lceil\ell/n\rceil$, so

$$|g'|\le\ell-\lceil\ell/n\rceil+1\le(1-1/n)|g|+1.\qquad\square$$

We shall modify the transducers $\Phi_{\mathbf{M}}$ by composing them with appropriate machines. We recall the general definition: let $\Phi\colon A\times S\to S\times A$ and $\Psi\colon B\times S\to S\times B$ be transducers with same stateset S. Their *composition* is the transducer $\Phi\circ\Psi$ with alphabet $A\times B$, given by

$$\begin{aligned}\Phi\circ\Psi\colon(A\times B)\times S=A\times(B\times S)&\xrightarrow{A\times\Psi}A\times(S\times B)\\&=(A\times S)\times B\xrightarrow{\Phi\times B}(S\times A)\times B=S\times(A\times B).\end{aligned}$$

We are given a transducer Φ with stateset $S=\{s_1,\dots,s_\ell,x,y\}$ and alphabet A. We write $G=\langle\Phi\rangle$, and freely identify words in S^* with their value in G. We require that x,y commute.

For every $i\in\{1,\dots,\ell\}$, consider the transducer Φ_i with alphabet $A_i=\{0,1\}$ and transitions $\Phi_i(a,q)=(a=0\,?\,q:\epsilon,q=s_i\,?\,1-a:a)$.

Note (by drawing the dual Moore diagram and deleting the transitions with ϵ output) that the only paths with input and output of same length are of the form $s_i^{-a}ws_i^b$ for some $a,b\in\{0,1\}$ and w a word not involving s_i.

Note also that for a word w of form $s_jx^my^n$

(1) if $i=j$ then $\Phi_i(0,w^2)=(w,0)$ and $\Phi_i(1,w^2)=(w',1)$ with w' conjugate to w;

(2) if $i\ne j$ then $\Phi_i(0,w)=(w,0)$ and $\Phi_i(1,w)=(\epsilon,1)$.

Consider also a transducer Φ_0 with alphabet $A_0 = \{0,1\}^3$ and transitions

$$\Phi_0((a,b,c),s_i) = (c = 0\,?\,s_i : \epsilon, (a,b,1-c)) \text{ for all } i;$$
$$\Phi_0((a,b,c),x) = (a = 0\,?\,x : \epsilon, (1-a,b,c));$$
$$\Phi_0((a,b,c),y) = (b = 0\,?\,y : \epsilon, (a,1-b,c)).$$

Note that, in the dual Moore diagram of Φ_0, all paths with input label of the form $s_i^{-a}x^m y^n s_j^b$ have shorter output label as soon as $|m| + |n| \geq 3$. Note also that if w is a word of the form $s_i x^m y^n$ then for all $(a,b,c) \in A_0$ we have $\Phi_0((a,b,c),w^2) = (w',(a,b,c))$ for some permutation w' of w; so in particular w' is conjugate to w because x and y commute. Furthermore, for $(a,b,c) = (0,0,0)$ we get $w' = w$ in G.

Proposition 3.6. *Under the hypotheses above, the transducer* $\Phi' := \Phi \circ \Phi_0 \circ \Phi_1 \circ \cdots \circ \Phi_\ell$ *generates a contracting group, and whenever we have* $\Phi(a,(s_i x^m y^n)^t) = (s_i' x^{m'} y^{n'}, a)$ *in the original transducer we have for all* $j \in \{0,1\}^{\ell+3}$ *the relation* $\Phi'((j,a),(s_i x^m y^n)^{4t}) = (w,(j,a))$, *with* w *either equal to* 1 *or conjugate to* $s_i' x^{m'} y^{n'}$. *Furthermore, if* $j = 0^{\ell+3}$ *then* $w = s_i' x^{m'} y^{n'}$.

Proof. After applying the transducers $\Phi_1, \ldots, \Phi_\ell$, the only words that don't get shortened are of the form $s_i^{-a} w(x,y) s_j^b$ for some $i,j \in \{1,\ldots,\ell\}$ and some $a,b \in \{0,1\}$. These get shortened by Φ_0 as soon as $|w| \geq 3$, using the fact that x and y commute. It follows that $\langle \Phi' \rangle$ is contracting.

Consider the transitions of $(s_i x^m y^n)^4$ in transducer $\Phi_1 \circ \cdots \circ \Phi_\ell$. On input letter 0^ℓ it produces $(s_i x^m y^n)^2$, on input letter $0\cdots1\cdots0$ with the '1' in position i it produces a conjugate of $(s_i x^m y^n)^2$ and on all other input letters it produces ϵ. Feed then $(s_i x^m y^n)^2$ to transducer Φ_0; on input letter 000 it produces $s_i x^m y^n$ and on all other input letters it produces a conjugate of $s_i x^m y^n$. Feed finally $s_i x^m y^n$ to Φ to conclude the proof. □

We are ready to finish the proof of Theorem 3.4. We constructed an integer-labeled graph for a transducer Φ, whose vertices are elements of G for Theorem B′ or symmetrized conjugacy classes for Theorem B.

By Proposition 3.6, the transducer Φ' is contracting. Let us check that the order problems for $\langle \Phi \rangle$ and for $\langle \Phi' \rangle$ are equivalent.

A graph for Φ' will have the same set of vertices as the graph for Φ, and Proposition 3.6 shows that this new graph has the same set of outgoing edges for each element of form $s_i x^m y^n$, with labels multiplied by 4 and, possibly, some new edges to 1 (or to $C(1)$). Since in the old graph there were no loops at non-identity elements, $s_* xy$ has infinite order in $\langle \Phi' \rangle$ if and only if it has infinite order in $\langle \Phi \rangle$, and the orbit of $(0,0^{\ell+3})^\infty$ is infinite under the action of $s_* xy \in \langle \Phi' \rangle$ if and only if the orbit of 0^∞ is infinite under the action of $s_* xy \in \langle \Phi \rangle$.

Finally, by replacing the stateset S by $\widetilde{S} = S \cup N$ and A by $\widetilde{A} = A^n$, we may assume that Φ' is nuclear.

4. Outlook

We proved in this article the undecidability of the order problem for automata groups, namely groups of transformations generated by a transducer.

If the transducer belongs to a restricted class, it may well be that the order problem becomes decidable. Klimann, Picantin and Savchuk compute in [16] orbits of automata groups and deduce some positive results on the order problem.

Here are some classes of transducers that seem to be of particular importance:

Transducers of polynomial growth. In a transducer Φ (represented by a graph as in Figure 1), let $\alpha(n)$ denote the number of paths of length n that end in a non-identity state. If $\alpha(n)$ is a bounded function (as is the case for the Grigorchuk group), then the order problem is solvable in $\langle\Phi\rangle$, see [5]. Is it still solvable if $\alpha(n)$ is bounded by a linear function? or by a polynomial of degree d? The groups generated by such transducers have been considered by Sidki [22].

Reset transducers. These are transducers Φ with $\Phi(a,s) = (\phi(a), \psi(a,s))$ for some functions ϕ, ψ; namely, the state reached by the transducer is independent of the original state. These transducers are intimately connected to tilings, by Kari's construction [14]. Gillibert proved in [7] that the order problem is unsolvable for semigroups of reset automata. Is it solvable in groups of reset automata?

Reversible transducers. These are transducers whose dual is invertible; they should be related to reversible Turing or Minsky machines. Is the order problem solvable for groups generated by reversible automata?

Bireversible transducers. These are transducers Φ such that all 8 transducers obtained from Φ by inverting or permuting the stateset and alphabet remain transducers; they give another point of view on square complexes (by tiling the plane with squares whose labels are (a, s, a', s') when $\Phi(a,s) = (s', a')$). Is the order problem solvable for groups generated by bireversible automata?

We expect it to be undecidable whether a functionally recursive group is actually an automata group (for a larger generating set), whether an automata group is contracting, and even whether a contracting group is finite (for the finiteness problem of automata groups, see [1]). Again, the related questions for semigroups are known to be undecidable by constructions in or similar to [7]. All these questions may be asked in terms of finiteness of orbits rather than finiteness of (cyclic) subgroups, as we actually did in this paper; see [2] for a general connection.

References

[1] A. Akhavi, I. Klimann, S. Lombardy, J. Mairesse, and M. Picantin, On the finiteness problem for automaton (semi)groups. *Internat. J. Algebra Comput.* **22** (2012), no. 6, 1250052, 26 pp. Zbl 1280.20038 MR 2974106

[2] D. D'Angeli, D. Francoeur, E. Rodaro, and J. Ph. Wächter, Infinite automaton semigroups and groups have infinite orbits. *J. Algebra* **553** (2020), 119–137. MR 4074167 Zbl 07181388

[3] J. M. Belk and C. Bleak, Some undecidability results for asynchronous transducers and the Brin–Thompson group $2V$. *Trans. Amer. Math. Soc.* **369** (2017), no. 5, 3157–3172. Zbl 1364.20015 MR 3605967

[4] B. Berger, The undecidability of the domino problem. *Mem. Amer. Math. Soc.* **66** (1966), 72 pp. Zbl 0199.30802 MR 0216954

[5] I. V. Bondarenko, N. V. Bondarenko, S. N. Sidki, and F. R. Zapata, On the conjugacy problem for finite-state automorphisms of regular rooted trees. *Groups Geom. Dyn.* **7** (2013), no. 2, 323–355. With an appendix by R. M. Jungers. Zbl 1286.20034 MR 3054572

[6] A. M. Brunner and S. N. Sidki, On the automorphism group of the one-rooted binary tree. *J. Algebra* **195** (1997), no. 2, 465–486. Zbl 0902.20017 MR 1469633

[7] P. Gillibert, The finiteness problem for automaton semigroups is undecidable. *Internat. J. Algebra Comput.* **24** (2014), no. 1, 1–9. Zbl 1292.20040 MR 3189662

[8] P. Gillibert, An automaton group with undecidable order and Engel problems. *J. Algebra* **497** (2018), 363–392. Zbl 1427.20040 MR 3743185

[9] R. I. Grigorchuk, On Burnside's problem on periodic groups. *Funktsional. Anal. i Prilozhen.* **14** (1980), no. 1, 53–54. In Russian. English translation, *Functional Anal. Appl.* **14** (1980), no. 1, 41–43. Zbl 0595.20029 MR 0565099

[10] R. I. Grigorchuk, On Milnor's problem of group growth. *Dokl. Akad. Nauk SSSR* **271** (1983), no. 1, 30–33. In Russian. English translation, *Soviet Math. Dokl.* **28** (1983), no. 1, 23–26. Zbl 0547.20025 MR 0712546

[11] R. I. Grigorchuk, V. V. Nekrashevych, and V. Suščans′kiĭ, Automata, dynamical systems, and groups. *Tr. Mat. Inst. Steklova* **231** (2000), Din. Sist., Avtom. i Beskon. Gruppy, 134–214. In Russian. English translation, *Proc. Steklov Inst. Math.* **2000,** no. 4(231), 128–203. Zbl 1155.37311 MR 1841755

[12] N. D. Gupta and S. N. Sidki, On the Burnside problem for periodic groups. *Math. Z.* **182** (1983), no. 3, 385–388. Zbl 0513.20024 MR 0696534

[13] Ju. Š. Gurevič, The problem of equality of words for certain classes of semigroups. *Algebra i Logika Sem.* **5** (1966), no. 5, 25–35. In Russian. Zbl 0178.32501 MR 0206079

[14] J. Kari, The nilpotency problem of one-dimensional cellular automata. *SIAM J. Comput.* **21** (1992), no. 3, 571–586. Zbl 0761.68067 MR 1163346

[15] I. Klimann, J. Mairesse, and M. Picantin, Implementing computations in automaton (semi)groups. In N. Moreira and R. Reis (eds.), *Implementation and application of automata.* Proceedings of the 17th Annual International Conference (CIAA 2012) held at the Universidade do Porto, Porto, July 17–20, 2012. Lecture Notes in Computer Science, 7381. Springer, Berlin etc., 2012, 240–252. Zbl 1297.68145 MR 2993189

[16] I. Klimann and M. Picantin, and D. Savchuk, Orbit automata as a new tool to attack the order problem in automaton groups. *J. Algebra* **445** (2016), 433–457. Zbl 1383.20018 MR 3418065

[17] V. Lukkarila, The 4-way deterministic tiling problem is undecidable. *Theoret. Comput. Sci.* **410** (2009), no. 16, 1516–1533. Zbl 1162.03024 MR 2502125

[18] M. L. Minsky, Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines. *Ann. of Math.* (2) **74** (1961), 437–455. Zbl 0105.00802 MR 0140405

[19] M. L. Minsky, *Computation*: *finite and infinite machines*. Prentice-Hall Series in Automatic Computation. Prentice-Hall, Englewood Cliffs, N.J., 1967. Zbl 0195.02402 MR 0356580

[20] V. V. Nekrashevych, *Self-similar groups*. Mathematical Surveys and Monographs, 117. American Mathematical Society, Providence, R.I., 2005. Zbl 1087.20032 MR 2162164

[21] R. Schroeppel, A two-counter machine cannot calculate 2^n. Massachusetts Institute of Technology. A. I. Laboratory, AIM-257. https://dspace.mit.edu/handle/1721.1/6202

[22] S. N. Sidki, Automorphisms of one-rooted trees: growth, circuit structure, and acyclicity. *J. Math. Sci.* (*New York*) **100** (2000), no. 1, 1925–1943. Zbl 1069.20504 MR 1774362

Received November 2, 2017

Laurent Bartholdi, Unité de Mathématiques Pures et Appliquées,
École Normale Supérieure de Lyon

Mathematisches Institut, Georg-August Universität zu Göttingen, Göttingen, Germany

e-mail: laurent.bartholdi@gmail.com

Ivan Mitrofanov, Département de Mathématiques, École Normale Supérieure, Paris, France

e-mail: phortim@yandex.ru