Ergod. Th. & Dynam. Sys., page 1 of 61

doi:10.1017/etds.2023.67

Distortion element in the automorphism group of a full shift

ANTONIN CALLARD† and VILLE SALO‡

† *GREYC, Université de Caen Normandie, 14000 Caen, France*
(e-mail: contact@acallard.net)
‡ *Department of Mathematics and Statistics, University of Turku, Turku, Finland*
(e-mail: vosalo@utu.fi)

(*Received* 31 *October* 2022 *and accepted in revised form* 17 *July* 2023)

Abstract. We show that there is a distortion element in a finitely generated subgroup G of the automorphism group of the full shift, namely an element of infinite order whose word norm grows polylogarithmically. As a corollary, we obtain a lower bound on the entropy dimension of any subshift containing a copy of G, and that a sofic shift's automorphism group contains a distortion element if and only if the sofic shift is uncountable. We obtain also that groups of Turing machines and the higher-dimensional Brin–Thompson groups mV admit distortion elements; in particular, $2V$ (unlike V) does not admit a proper action on a CAT(0) cube complex. In each case, the distortion element roughly corresponds to the SMART machine of Cassaigne, Ollinger, and Torres-Avilés [A small minimal aperiodic reversible Turing machine. *J. Comput. System Sci.* **84** (2017), 288–301].

Key words: group distortion, automorphism groups, cellular automata, subshifts
2020 Mathematics Subject Classification: 37B15 (Primary); 20F65 (Secondary)

1. *Introduction*

We begin with an introduction to automorphism groups and the topic of distortion in §1.1, as this is the motivation and context for our main results listed in §1.2.

The proofs of the main results are based on rather different ideas, namely conveyor belts, 'ducking', dynamics of Turing machines, permutation groups, and also some ideas from computer science, namely reversible computation and logical gates. Some background for these ideas is given in §1.3.

1.1. *Automorphism groups and distortion.*

A recent trend in symbolic dynamics is the study of automorphism groups of subshifts. Typical activities include the study of

restrictions that dynamical properties of the subshift put on these groups, and in turn constructing complicated automorphism groups or subgroups thereof.

The former activity has been most successful in the low-complexity setting, see [43] for a recent account of the state of the art. For example, minimal subshifts with upper entropy dimension less than $1/2$ have amenable automorphism groups [18], and (as discussed in more depth below) zero-entropy subshifts do not admit elements with exponential distortion [17].

The latter activity has been most successful on sofic shifts. In particular, a lot is known about the finitely generated subgroups of automorphism groups of full shifts: see [47] for a listing of properties that have been exhibited. For instance, let us mention that these groups G, while not finitely generated, contain finitely generated 'f.g.-universal subgroups', namely ones that contain isomorphic copies of all finitely generated subgroups of G. The class of subgroups of automorphism groups is also quite robust, being closed under graph products [46, 47]. A classical reference for the study of automorphism groups of transitive SFTs (an important subclass of sofic shifts) is [10].

In this paper, we study the group-theoretic notion of distortion, introduced by Gromov [26], in the context of automorphism groups of subshifts. If G is a finitely generated group, we say $g \in G$ is a *distortion element*, or *distorted*, if g is of infinite order and the word norm $|g^n|$ grows sublinearly (with respect to some, or equivalently any, finite generating set). For groups that are not finitely generated, we say that an element is distorted if it is distorted in some finitely generated subgroup. While distortion elements are usually allowed to have finite order, in this paper, we focus on distorted elements of infinite order.

Two basic examples of groups with distortion elements are the Heisenberg group with presentation $\langle a, b \mid [[a, b], a], [[a, b], b]\rangle$, where the element $[a, b]$ has quadratic distortion, meaning we can represent an element of the form $[a, b]^{\Omega(n^2)}$ by composing n generators; and the Baumslag–Solitar group BS(1, 2) with presentation $\langle a, b \mid a^b = a^2\rangle$, where a is easily seen to be exponentially distorted, meaning the word norm of a^n grows logarithmically.

The previous examples show that distortion elements can appear in nilpotent and metabelian linear groups. It is known that they cannot appear in biautomatic groups [25], certain types of mapping class groups [21], and the outer automorphism group of the free group [1]. See [12, 13, 23, 24, 27, 36, 40, 44] for other distortion-related works.

Getting back to automorphism groups, it is an open problem (that we solve in the present paper) whether the automorphism group of any subshift can contain a distortion element [17]. It is not known whether the Heisenberg group [33] or the Baumslag–Solitar group BS(1, 2) embed in $\mathrm{Aut}(A^{\mathbb{Z}})$, or indeed in the automorphism group of any subshift, and these problems stay open. (It is also open whether the additive group of dyadic rationals $\mathbb{Z}[\frac{1}{2}] \leq \mathrm{BS}(1, 2)$ embeds in $\mathrm{Aut}(A^{\mathbb{Z}})$ [10].)

In addition to being an interesting group-theoretic notion, the quest for distortion elements in automorphism groups of subshifts is motivated by several purely symbolic dynamical considerations. First, [18, Theorem 1.2] shows that finitely generated torsion-free subgroups of the automorphism group of a subshift of polynomial complexity are virtually nilpotent. See [20, Theorem 5.5] for a similar conclusion for inverse limits

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

of bounded step nilsystems. If we could rule out distortion in such examples, we could conclude virtual abelianness.

Second, it is known that the Baumslag–Solitar group, more generally any group with an exponentially distorted element, does not embed in the automorphism group of a zero-entropy subshift [17]. More precisely, it was observed there that the Morse–Hedlund theorem allows one to translate a distortion element into a lower bound on the complexity of a subshift. This is notable, as this is the only known restriction for automorphism groups of general zero-entropy subshifts. Thus, distortion looks like a natural candidate for restrictions on automorphism groups of general subshifts (as far as the authors know, no restrictions are known on countable subgroups of automorphism groups of general subshifts).

Third, distortion is tied to an intrinsic notion in automorphism group theory, namely the growth of the *radius* (as known as range) of the automorphism, when seen as a cellular automaton. Namely, distortion in the group sense implies sublinear growth of the radius [16]. It is not immediately obvious that even sublinear radius growth is possible (indeed this was left open in [16]), but several examples of sublinear radius growth have been constructed. The most relevant for us is the observation from [28] that one can even obtain sublinear radius growth in the automorphism group of a full shift: the so-called *SMART machine*, when simulated by an automorphism, gives rise to such growth.

While distortion elements have not previously been exhibited in automorphism groups of subshifts, some facts are known about their dynamics (mostly related the notion of radius). Links to expansive directions and Lyapunov exponents are shown in [16]. A related result is shown in [6], namely distortion elements of automorphism groups of general expansive systems can not themselves be expansive. Links to the dimension group action and inertness are discussed in [16, 50].

1.2. *Results.* The main result of the present paper is that the automorphism group of some full shift (thus any full shift by standard embedding theorems [33]) contains a distortion element with 'quasi-exponential' distortion, in the sense that the distortion function grows like $\exp(\sqrt[4]{\Omega(n)})$. It is more convenient to work directly with word norms than with the distortion function, so we take this approach in the paper. Note that for well-behaved functions, the word norm growth is just the inverse of the distortion function.

THEOREM A. *For any non-trivial alphabet A, the group $\mathrm{Aut}(A^{\mathbb{Z}})$ has an element g of infinite order such that $|g^n|_F = O(\log^4 n)$ for some finite set F.*

Here, by a *non-trivial alphabet*, we mean a finite set A with $2 \leq |A| < \infty$; we also use the standard shorthand $\log^4 n = (\log n)^4$.

A simple counting argument shows that the word norms of nth powers of a group element cannot be $o(\log n)$ with respect to a fixed finite generating set. For our specific automorphism, one can strengthen this: the radius of g^n as a cellular automaton is $\Theta(\log n)$, so the true growth of word norms of powers of our automorphism is between $\Omega(\log n)$ and $O(\log^4 n)$.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Our theorem solves the second subquestion of [17, Question 5.1] in the affirmative. Most of the present paper deals with the proof of this theorem. The element g in this theorem is essentially the SMART machine [14], so morally this also confirms a conjecture of [28], although the embedding we use is slightly more involved than the specific one considered in [28]. The group we use in the proof is given in Lemma 5.14.

The generators of our group are relatively simple, but we have little idea what kind of group they generate. The finitely generated group $\langle F \rangle$ can of course be taken to be larger (the distortion function can only become faster-growing this way), so one can take a more canonical choice of generators, say, all reversible cellular automata with biradius 1 (on the huge alphabet we use).

One can perform some further massage to get a simpler-sounding example: it is known that the automorphism group of a full shift contains so-called finitely generated (f.g.)-universal subgroups, namely ones containing copies of all f.g. groups of reversible cellular automata [48]. Any such group can be used in the result (although the element g will be more complicated). In particular, one can pick as F the symbol permutations and the partial shift $\sigma \times \mathrm{id}$ on the product full shift $\{0,1\}^{\mathbb{Z}} \times \{0,1,2\}^{\mathbb{Z}}$.

From the main theorem, we obtain several corollaries of interest, which are proved in §6. First, we obtain the characterization of the class of sofic shifts whose automorphism groups have distortion elements.

THEOREM B. *Let X be a sofic shift. Then $\mathrm{Aut}(X)$ contains a distortion element if and only if X is uncountable.*

It is well known that for sofic shifts, uncountability is equivalent to having positive entropy.

As another immediate consequence, using the argument of [17], we obtain that the automorphism group of a full shift cannot be embedded in the automorphism group of a low-complexity subshift. Recall that the *lower entropy dimension* [37] of a subshift is defined by the formula

$$\underline{D}(X) = \liminf_{k \to \infty} \frac{\log(\log N_k(X))}{\log k},$$

where $N_k(X)$ is the number of words of length k that appear in X. The lower entropy dimension of a (one-dimensional) subshift with positive entropy is of course 1. The *upper entropy dimension* is defined analogously, with lim sup in place of lim inf.

LEMMA 1.1. *Let X be a subshift with lower entropy dimension less than $1/d$. If $f \in \mathrm{Aut}(X)$ satisfies $|f^n| = O(\log^d n)$, then f is periodic.*

THEOREM C. *The group $\mathrm{Aut}(A^{\mathbb{Z}})$ has a finitely generated subgroup G such that every subshift X with $G \leq \mathrm{Aut}(X)$ has lower entropy dimension at least $1/4$.*

Theorem C is of course an immediate corollary of Lemma 1.1. It states a *low-complexity restriction* on the automorphism group, that is, it states that automorphism groups of subshifts with low enough complexity (growth of the number of admissible words) cannot

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

have some property. The above theorem seems to be the first low-complexity restriction on automorphism groups where:

(1) the complexity bound is superpolynomial;
(2) there are no additional dynamical restrictions; and
(3) the prevented behavior can be exhibited in the automorphism group of another subshift.

There are previously known restrictions satisfying any two of these items. For items (1) and (2), zero entropy prevents exponential distortion [17]; for items (1) and (3), [18] shows that if X is minimal and has upper entropy dimension less than $1/2$, then it is amenable (while $\mathrm{Aut}(A^{\mathbb{Z}})$ is not); for items (2) and (3) (very low complexity restrictions), there are many results, see [43].

The subgroup where our distortion element lies can itself be seen as a group of Turing machines, indeed restricting its action to a certain sofic subshift directly gives rise to a subgroup of the group $\mathrm{RTM}(n, k)$ studied in [3], leading to the following theorem.

THEOREM D. *Let $n \geq 2$, $k \geq 1$. Then the group of Turing machines $\mathrm{RTM}(n, k)$ contains a distortion element; indeed there is a finitely generated subgroup $G = \langle F \rangle$ and an element f such that $|f^n|_F = O(\log^4 n)$.*

All groups of Turing machines in turn embed in the higher-dimensional Brin–Thompson mV for $m \geq 2$ introduced by Brin [11], and we obtain the following theorem.

THEOREM E. *The Brin–Thompson group mV contains a distortion element; indeed there is an element f such that $|f^n| = O(\log^4 n)$.*

This theorem provides a new restriction for geometries of $2V$. Namely, it is known that Thompson's group V admits a proper action by isometries on a CAT(0) cube complex [22]. By [29, Theorem 1.5], a group with distortion elements does not admit such an action, thus we have the following corollary.

COROLLARY 1.2. *The Brin–Thompson group mV does not act properly on a CAT(0) cube complex for $m \geq 2$.*

Of course, a similar fact is true for the other groups where we exhibit distortion elements.

We conclude with previously known (but possibly not well known) related distortion facts that are easy to prove. First, the fact the automorphism group of a full shift contains finitely generated *subgroups* that are distorted is essentially classical, namely $F_2 \times F_2$ embeds in $\mathrm{Aut}(A^{\mathbb{Z}})$ [33] and has subgroups with arbitrarily bad (recursive) distortion essentially by [38]. (Given a subgroup H and an overgroup G equipped with their respective word norms, the subgroup H is distorted in G if $\min\{\|h\|_G \mid h \in H \text{ and } \|h\|_H \geq n\} = o(n)$. In this sense, a distorted element corresponds to a distorted cyclic subgroup.) To give a more down-to-earth example, $\mathbb{Z}_2 \wr \mathbb{Z}^2$, which embeds in $\mathrm{Aut}(A^{\mathbb{Z}})$ by [47], contains a polynomially distorted copy of itself, by a nice geometric argument [19]. One can also construct distorted subgroups directly by more intrinsic automorphism group techniques.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Second, in the setting of general expansive homeomorphisms, finding distortion elements is very easy. Namely, if $\mathbb{S}$ is the invertible natural extension of the $\times 2$-map on the circle, $\mathrm{Aut}(\mathbb{S}^n)$ contains a natural copy of $\mathrm{GL}(n, \mathbb{Z})$ by simply summing tracks to each other [34]. For $n = 3$, the group $\mathrm{GL}(n, \mathbb{Z})$ contains the Heisenberg group, thus has distortion elements.

1.3. *Turing machines and gates.* While our results in the previous section are stated fully in terms of homeomorphism groups, our *proof methods* rather belong to the theories of dynamical Turing machines and of reversible gates. In this section, we outline some history of these ideas.

1.3.1. *Turing machines.* As mentioned in §1.1, our automorphism group element simulates a 'Turing machine', that is, a dynamical system where a single head moves over an infinite tape of arbitrary data (over a fixed finite alphabet), and all the action happens near the head (which may move around the tape, such movement depending on the content of the tape; or modify said content). The dynamics of Turing machines, also known as one-head machines, is an important branch of symbolic dynamics. This can be seen as initiated in the 1997 paper of Kůrka [35], which explicitly defined the moving-head and moving-tape dynamics of Turing machines (although many relevant dynamical ideas appeared in the literature before this [30, 39, 45]).

One of the most-studied behaviors of Turing machines is aperiodicity, meaning that the action of the Turing machine has no periodic points. This property is particularly interesting in the moving tape model, where the head is seen as fixed and only the tape moves. Kůrka originally conjectured that Turing machines cannot be aperiodic, but an explicit aperiodic Turing machine was exhibited in 2002 by Blondel, Cassaige, and Nichitiu [7] (inspired by a technique of Hooper from 1966 [30]). Later, reversible aperiodic Turing machines (ones whose action is a homeomorphism) were found, the first by Kari and Ollinger [32]. This culminated in the discovery of the SMART machine $\mathcal{S}$ by Cassaigne, Ollinger, and Torres-Avilés [14], a machine with only four states and three tape-letters, which is reversible and aperiodic, and whose moving-tape dynamics is a minimal homeomorphism on the Cantor space, see also [41].

Turing machines, in the moving-head dynamics where the tape is not shifted and the head moves over it, can be directly seen as automorphisms of a sofic shift [3]. In fact, it is well known that Turing machines can be 'embedded' into automorphism groups of full shifts $\mathrm{Aut}(A^{\mathbb{Z}})$. There are multiple ways of doing so; in this paper, we use the conveyor belt technique similar to the one used in [28].

For the purpose of establishing distortion, the first important consideration, already discussed in §1.1, is the 'speed' of a Turing machine: a Turing machine with positive speed, meaning the existence of tape contents such that the head moves to infinity at a positive rate, could not possibly give rise to a distortion element. This is because the linear movement of the head (even on a single configuration) means that the radius of powers of the corresponding automorphism must grow at a linear rate as well, which prevents distortion.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

It was shown in [31] that all aperiodic Turing machines have zero speed, and in [28], this was strengthened by proving that the maximal offset by which such a machine can move in t time steps is $O(t/\log t)$. For the SMART machine $\mathcal{S}$, more is known: in t steps, it can only move by an offset of at most $O(\log t)$. This makes $\mathcal{S}$ a perfect candidate for a distortion element of a subshift automorphism group, and indeed it was conjectured in [28] that it is one.

1.3.2. *Gates.* The next ideas come from the study of reversible gates. By this, we refer to the study of permutation groups acting on (a sublanguage of) A^n, where A is a finite alphabet, which are generated by 'reversible gates': that is, permutations that only consider a bounded subset of coordinates at a time. More precisely, if $k \leq n$ and $\pi \in \mathrm{Sym}(A^k)$, then we can apply π to the subword starting at i by the formula $\hat{\pi}(u \cdot v \cdot w) = u \cdot \pi(v) \cdot w$, where $u \in A^i$, $v \in A^k$, $w \in A^{n-k-i}$. From now on, we use the term 'gate' for reversible gates and 'classical gate' to refer to the usual not necessarily reversible gates (in the few places where they are needed).

A fundamental lemma in this topic is that $\mathrm{Alt}(A^n)$ admits a generating set with bounded k, namely it is generated by the even permutations of A^2 if the cardinality $\#A$ is at least 3 (when we consider them as gates, and allow their applications at any position $i = 0, \ldots, n-2$). A more complete statement appears in [48], while earlier proofs are given in [8, 9, 51].

The connection between gates and Turing machines is as follows. Let us consider generalized Turing machines in the sense of [3], meaning the machine can look at and modify multiple cells at once, although only at a bounded distance from the head. Now, walking on a cyclic tape containing an element of A^n, we can apply permutations of A^k at different relative positions i: simply move by i steps, apply the permutation locally, and then move back by $-i$ steps. The above paragraph translates to the fact that there is a finite set of generalized Turing machines that can perform any even permutation of the tape content (relative to the head position). Actually, it turns out that since Turing machines carry a state, $k = 1$ suffices, that is, the generating Turing machines need not be of a generalized type.

2. *Definitions*

2.1. *General notions.* We have $\mathbb{N} = \{0, 1, 2, \ldots\}$, $\mathbb{Z}_+ = \mathbb{N} \setminus \{0\}$, and $\mathbb{Z}_\ell = \mathbb{Z}/\ell\mathbb{Z}$ is integers modulo ℓ. For S a finite set, we denote by $\#S$ the cardinality of S. For $i, j \in \mathbb{N}$, denote $[\![i, j]\!] = \{n \in \mathbb{N} : i \leq n \leq j\}$ and $[\![n]\!] = [\![0, n-1]\!]$. If $w \in \{0, 1, \ldots, k-1\}^*$, write $\mathrm{v}_k(w)$ for the value w represents in base k (the leftmost digit having the highest significance by default), that is, $\mathrm{v}_k(w) = \sum_{i=0}^{|w|-1} k^{|w|-1-i} w_i$; and we write $n_{(k)} \in \{0, \ldots, k-1\}^*$ for the number $n \in \mathbb{N}$ written in base k (with length determined from context or specified in text), that is, $\mathrm{v}_k(n_{(k)}) = n$.

For Σ a finite set, called an *alphabet*, denote by $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$ the set of finite words over Σ. For $w \in \Sigma^*$, denote by $\mathrm{len}(w)$ the length of w, that is, the integer n such that $w \in \Sigma^n$. For a word $w \in \Sigma^*$, denote by $\overline{w}$ the reverse (or 'mirror image') of w, that is, if $w = w_0 \cdot w_1 \cdots w_{n-1}$, then $\overline{w} = w_{n-1} \cdot w_{n-2} \cdots w_0$. For $w \in \Sigma^n$ and

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

$J \subseteq [\![0, n-1]\!]$, define $w|_J = w_{j_0} \cdot w_{j_1} \cdots w_{j_k}$ as the restriction of w to J, for $J = \{j_0, \dots, j_k\}$ and $j_0 \leq \cdots \leq j_k$. Given $a \in \Sigma$ and $j \in [\![0, n-1]\!]$, the cylinder $[a]_j$ denotes the set of words $\{w \in \Sigma^n \mid w_j = a\}$. Usually our alphabets are *non-trivial*, by which we mean $|\Sigma| \geq 2$.

In Lemma 4.10, we denote NC^1 for 'Nick's Class' of complexity of level 1, that is, the class of languages $L \subseteq \Sigma^*$ such that L is decidable by Boolean circuits with a polynomial number of gates, with at most two inputs and depth $O(\log n)$ (see for example [2]). The reader need not be familiar with this class to follow our argument. The main technical result we need is Barrington's theorem from [4] (but this is also proved from scratch in our context).

For a, b elements of a group, the commutator of a and b is $[a, b] = a^{-1}b^{-1}ab$. The conjugation convention is $a^b = b^{-1} \circ a \circ b$. If $\pi \in \mathrm{Sym}(A)$ is a permutation, we may regard it as a permutation of $A \times B$ by $\pi((a, b)) = (\pi(a), b)$. Our groups always act from the left. If $g_1, \dots, g_n$ are commuting elements of a group, we write $\prod_{i=1}^{n} g_i$ for their ordered product $g_n \cdots g_1$. In groups of bijections on a set (which almost all our groups are), we denote composition by $\circ$.

Given a finitely generated group G generated by the finite set S, a *presentation* of $g \in G$ is a word $w = s_n \cdots s_1 \in (S \cup S^{-1})^*$ such that $g = s_n \cdots s_1$, and we write $w \equiv g$. The *word norm* $\|g\|_S$ *of* $g \in G$ *relative to* S is then the length of a shortest presentation of g, that is, $\|g\|_S = \min\{n \in \mathbb{N} : \text{ there exists } w \in (S \cup S^{-1})^n, w \equiv g\}$. This word norm is also the distance in the Cayley graph of G between 1_G and g. In this context, an element $g \in G$ is said to be *distorted* if $\|g^n\|_S = o(n)$.

For sets X, Y, and Z, we say that a map $f : X \to Y$ *lifts* into $\tilde{f} : X \times Z \to Y \times Z$ (or that $\tilde{f}$ is the *lift* of f) if $\tilde{f}(x, z) = (f(x), z)$. For $S \subseteq X$ a subset of X, and $f : X \to X$, we call *extended restriction* of f to S the map $f_{/S} : X \to X$ defined as

$$f_{/S}(x) = \begin{cases} f(x) & \text{if } x \in S, \\ x & \text{otherwise,} \end{cases}$$

that is, $f_{/S}$ is the extension of the restriction $f|_S$ back to the full domain X, by fixing elements outside S.

2.2. *Subshifts and cellular automata.* Let Σ be a finite alphabet. An element $x \in \Sigma^{\mathbb{Z}}$ is called a *configuration*. An element $w \in \Sigma^*$ is called a *word* or a *pattern*, and a pattern $w \in \Sigma^*$ is said to *appear* in a configuration $x \in \Sigma^{\mathbb{Z}}$, denoted $w \sqsubseteq x$ if there exists some $i \in \mathbb{Z}$ such that $x_{i+j} = w_j$ for every $j \in [\![0, \mathrm{len}(w) - 1]\!]$.

We endow $\Sigma^{\mathbb{Z}}$ with the product topology. This topology is generated by the cylinders $[a]_j = \{x \in \Sigma^{\mathbb{Z}} : x_j = a\}$ for $a \in \Sigma$ and $j \in \mathbb{Z}$. The *left shift* $\sigma : \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$ defined by $\sigma(x)_i = x_{i+1}$ is a $\mathbb{Z}$ action on $\Sigma^{\mathbb{Z}}$. Closed and shift-invariant subsets X of $\Sigma^{\mathbb{Z}}$ are called *subshifts*. For X a subshift and $n \in \mathbb{N}$, we denote by $\mathcal{L}_n(X)$ the set of finite words of length n that appear in X, and by $\mathcal{L}(X) = \bigcup_{n \in \mathbb{N}} \mathcal{L}_n(X)$ its *language*. We say that a subshift X is *sofic* if $\mathcal{L}(X)$ is a regular language. If X and Y are subshifts, a continuous and shift-equivariant map $f : X \to Y$ is called a *morphism*. It is an *endomorphism* if $X = Y$ and an *automorphism* if, in addition,

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

it is bijective (in which case f^{-1} is also an endomorphism). Endomorphisms are sometimes called *cellular automata*, and automorphisms *reversible cellular automata*. For $f : X \to Y$ a morphism between two subshifts, its *radius* (as a cellular automaton) is the minimal r such that $f(x)_i$ is a function of $x_{[\![i-r,i+r]\!]}$. The *biradius* of an automorphism is the maximum of the radii of f and f^{-1}.

2.3. *Turing machines.* In this article, we use Turing machines as a specific kind of action on subshifts. We note that despite the terminology, it is not necessarily helpful to think of them as computational devices. (We will later perform computation in a group of Turing machines, but this computation is not related to the usual type of Turing machine computation, in that the iteration of a single machine is *not* going to be used to perform computation.)

Let Q be a finite set called the *state set*, and Γ be a finite set called the *tape alphabet*. In the model of [32] (this model is equivalent to the usual definition of Turing machines, but handles reversibility better), a *Turing machine* is a triple $\mathcal{M} = (\Gamma, Q, \Delta)$, where $\Delta \subseteq (Q \times \{+1, -1\} \times Q) \cup (Q \times \Gamma \times Q \times \Gamma)$ is the *transition table*. A transition $(q, \delta, q') \in Q \times \{+1, -1\} \times Q$ is called a *move transition*, and a transition $(q, a, q', b) \in Q \times \Gamma \times Q \times \Gamma$ is called a *matching transition*.

In the rest of this paper, we focus on the action of Turing machines on two families of objects: bi-infinite tapes and finite cyclic tapes.

2.3.1. *Bi-infinite tapes.* In the alphabet $\Gamma \cup (Q \times \Gamma)$, elements of $H = Q \times \Gamma$ are called *heads*. Denote by

$$\mathcal{X} = \left\{ x \in (\Gamma \cup (Q \times \Gamma))^{\mathbb{Z}} \mid \text{ for all } i, j \in \mathbb{Z} : i \neq j \implies x_i \in \Gamma \vee x_j \in \Gamma \right\}$$

the set of bi-infinite tapes with at most one head somewhere. We can associate to $\mathcal{M}$ its so-called *moving-head model* [35], that is, the binary relation $\to_{\mathcal{M}}$ on $\mathcal{X}$ defined by $x \to_{\mathcal{M}} x$ if $x \in \mathcal{X}$ contains no head (that is, $x \in \Gamma^{\mathbb{Z}}$); and if $x \in \mathcal{X}$ contains a head at position, say $i_0 \in \mathbb{Z}$ with $x_{i_0} = (q, a)$ for some $q \in Q$ and $a \in \Gamma$, then $x \to_{\mathcal{M}} x'$ if there exists $t \in \Delta$ such that:

$$\text{if } t = (q, a, q', b) \in \Delta : \qquad x'_i = \begin{cases} (q', b) & \text{if } i = i_0, \\ x_j & \text{otherwise;} \end{cases}$$

$$\text{if } t = (q, \delta, q') \in \Delta : \qquad x'_i = \begin{cases} a & \text{if } i = i_0, \\ (q', x_i) & \text{if } i = i_0 + \delta, \\ x_i & \text{otherwise.} \end{cases}$$

The binary relation $\to_{\mathcal{M}}$ on $\mathcal{X}$ (denoted $\to$ for short if the context is clear) is the *reachability* relation. We write $\to^k_{\mathcal{M}}$ its kth power, and $\to^*_{\mathcal{M}}$ its transitive closure. We say that $\mathcal{M}$ reaches the configuration x' from x in $k \in \mathbb{N}$ steps if $x \to^k_{\mathcal{M}} x'$. A transition $x \to^*_{\mathcal{M}} x'$ is called a *move*.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

The machine $\mathcal{M}$ is *deterministic* if $\to_{\mathcal{M}}$ defines a partial function, *complete deterministic* if it defines a total function (which is then continuous and, obviously, shift-commuting), and *complete reversible* (or *reversible* for short) if it defines a bijection (which is then a homeomorphism). When $\mathcal{M}$ is complete deterministic (which all our machines are), when using the relation $\to_{\mathcal{M}}$ as a function, we write it as $T_{\mathcal{M}} : \mathcal{X} \to \mathcal{X}$, which is an endomorphism of the subshift $\mathcal{X}$. Similarly, when the machine $\mathcal{M}$ is reversible, it is an automorphism of $\mathcal{X}$.

2.3.2. *Finite cyclic tapes.* The set of *cyclic configurations of length* $\ell \in \mathbb{Z}_+$ is the set

$$C_\ell = \left\{ x \in (\Gamma \cup (Q \times \Gamma))^{\mathbb{Z}/\ell\mathbb{Z}} \mid \text{for all } i, j \in \mathbb{Z}/\ell\mathbb{Z}, i \neq j \implies x_i \in \Gamma \vee x_j \in \Gamma \right\}$$

of finite configurations containing at most one head. We always assume $\ell \geq 2$ in what follows (the case $\ell = 1$ makes sense, but requires notational modifications and is the least interesting case anyway).

The machine $\mathcal{M}$ defines a binary relation $\to_{\mathcal{M}}$ on C_ℓ by considering these finite tapes as cyclic, that is, we define $x \to_{\mathcal{M}} x$ if $x \in C_\ell$ contains no head (that is, $x \in \Gamma^{\mathbb{Z}/\ell\mathbb{Z}}$); and if $x \in C_\ell$ contains a head at position, say $i_0 \in \mathbb{Z}/\ell\mathbb{Z}$ with $x_{i_0} = (q, a)$ for some $q \in Q$ and $a \in \Gamma$, then $x \to_{\mathcal{M}} x'$ if there exists $t \in \Delta$ such that:

$$\text{if } t = (q, a, q', b) \in \Delta : \qquad x'_i = \begin{cases} (q', b) & \text{if } i = i_0, \\ x_i & \text{otherwise;} \end{cases}$$

$$\text{if } t = (q, \delta, q') \in \Delta : \qquad x'_i = \begin{cases} a & \text{if } i = i_0, \\ (q', x_i) & \text{if } i = i_0 + \delta \bmod \ell, \\ x_i & \text{otherwise.} \end{cases}$$

As above, the relation $\to_{\mathcal{M}}$ is called the *reachability* relation. If $\mathcal{M}$ is complete deterministic, the function $\to_{\mathcal{M}}$ will be denoted by $T_{\ell,\mathcal{M}} : C_\ell \to C_\ell$. Note that it is an endomorphism of the shift action of $\mathbb{Z}$ (or $\mathbb{Z}_\ell$) which translates the cyclic tape around.

Finally, for any machine $\mathcal{M} = (Q, \Gamma, \Delta)$, denote by $m : \mathbb{N} \to \mathbb{N}$ its *movement function*, that is, $m(n)$ is the maximal number of cells the machine can visit in n steps. More precisely, $m(n)$ is the length $r - l + 1$ of the largest interval $[\![l, r]\!] \subseteq \mathbb{Z}$ such that there exists a sequence of n steps of computation $x_0 \to_{\mathcal{M}} x_1 \to_{\mathcal{M}} \cdots \to_{\mathcal{M}} x_n$ (with $x_i \in \mathcal{X}$) such that for every position $i \in [\![l, r]\!]$, at least one of the tapes x_k ($0 \leq k \leq n$) has its head at position i.

3. *The SMART machine on cyclic tapes*

Let SMART be the Turing machine (Q, Γ, Δ), where $Q = \{\blacktriangleright_1, \blacktriangleleft_1, \triangleright_1, \triangleleft_1\} \cup \{\blacktriangleright_2, \blacktriangleleft_2, \triangleright_2, \triangleleft_2\}$, $\Gamma = \{0, 1, 2\}$ and Δ is the transition table as shown in Figure 1.

We refer to $\blacktriangleright_1$, $\blacktriangleright_2$, $\blacktriangleleft_1$, $\blacktriangleleft_2$ (respectively $\triangleright_1$, $\triangleright_2$, $\triangleleft_1$, $\triangleleft_2$) as *filled* (respectively *hollow*) *triangles*.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

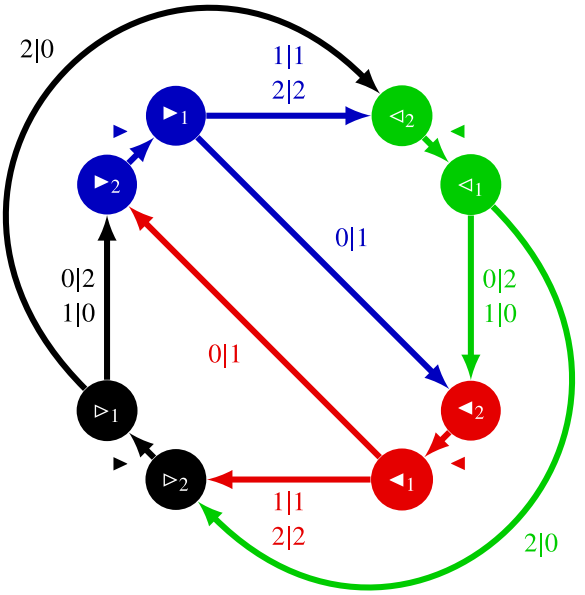


FIGURE 1. An arrow from q to q' labeled $\blacktriangleright$ (respectively $\blacktriangleleft$) denotes a transition $(q, +1, q')$ (respectively $(q, -1, q')$). An arrow from q to q' labeled $a|b$ denotes a transition (q, a, q', b).

Remark 3.1. The SMART machine was introduced with a slightly different formalism in [14], and slightly revised in [41] (states were renamed and permuted). The machine above adapts the latter in the model of [32] for Turing machines: in other words, we duplicate the states. We kindly advise readers already familiar with the SMART machine to read these definitions and propositions carefully.

Namely, while our SMART machine is in a sense completely equivalent, in the formulas in Proposition 3.2 describing traversals of SMART over zeroes, the patterns corresponding to filled and hollow initial states are of the same length (unlike the corresponding ones in [14]). This will be helpful later, when we encode the position in the sweep into the corresponding area on the tape without any extra space.

In this section, we consider the action of this machine on finite patterns (denoted with rounds brackets) like

$$\begin{pmatrix} 1 & 1 & 0^k & 2 \\ \blacktriangleright_2 & & & \end{pmatrix}$$

The argument applies whether or not these are finite subpatterns of a finite cyclic tape, or of an infinite configuration. When specifying a move (with some number of transition steps) between two patterns, it is implicit that the initial and final patterns have the same domain, and the machine does not exit this domain during the intermediate steps. Complete cyclic configurations (where the notation specifies the contents of all ℓ cells) will be denoted similarly, but with square brackets.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

PROPOSITION 3.2. (Adapted from [14, Lemma 1]) *Let $f(k) = 3^{k+1} - 2$. For all k, $s_* \in \{0, 1, 2\}$, and $s_+ \in \{1, 2\}$, the following moves hold:*

$$M_{\blacktriangleright}(k) : \begin{pmatrix} s_+ & 0^k & s_* \\ \blacktriangleright_2 & & \end{pmatrix} \to^{f(k)} \begin{pmatrix} s_+ & 0^k & s_* \\ & & \blacktriangleright_1 \end{pmatrix} \quad M_{\blacktriangleleft}(k) : \begin{pmatrix} s_* & 0^k & s_+ \\ & & \blacktriangleleft_2 \end{pmatrix} \to^{f(k)} \begin{pmatrix} s_* & 0^k & s_+ \\ \blacktriangleleft_1 & & \end{pmatrix}$$

$$M_{\triangleright}(k) : \begin{pmatrix} s_* & 0^k & s_+ \\ \triangleright_2 & & \end{pmatrix} \to^{f(k)} \begin{pmatrix} s_* & 0^k & s_+ \\ & & \triangleright_1 \end{pmatrix} \quad M_{\triangleleft}(k) : \begin{pmatrix} s_+ & 0^k & s_* \\ & & \triangleleft_2 \end{pmatrix} \to^{f(k)} \begin{pmatrix} s_+ & 0^k & s_* \\ \triangleleft_1 & & \end{pmatrix}$$

Additionally, the cell containing s_* is only visited at the last (respectively first) step of the sequences of transitions $M_{\blacktriangleright}$ and $M_{\blacktriangleleft}$ (respectively $M_{\triangleright}$ and $M_{\triangleleft}$). And the cell containing s_+ is never modified.

Proof. This proof adapts the proof of [14, Lemma 1], and highlights the recursive/nested aspects of these moves. In the case $k = 0$, one can check that indeed the formula describes a single transition. We reason by induction, and assume $M_{\blacktriangleright}(k)$, $M_{\blacktriangleleft}(k)$, $M_{\triangleright}(k)$, and $M_{\triangleleft}(k)$ hold. We only prove $M_{\blacktriangleright}(k+1)$ and $M_{\triangleright}(k+1)$, by symmetry between $\blacktriangleright$ and $\blacktriangleleft$ (respectively $\triangleright$ and $\triangleleft$). Since $f(k+1) = 3f(k) + 4$, we should find 3 recursions and 4 extra steps. This is what happens:

$M_{\blacktriangleright}(k+1)$		$M_{\triangleright}(k+1)$	
	$\begin{pmatrix} s_+ & 0^k & 0 & s_* \\ \blacktriangleright_2 & & & \end{pmatrix}$		$\begin{pmatrix} s_* & 0 & 0^k & s_+ \\ \triangleright_2 & & & \end{pmatrix}$
	Apply $M_{\blacktriangleright}(k)$		Apply one step
$\to^{f(k)}$	$\begin{pmatrix} s_+ & 0^k & 0 & s_* \\ & & \blacktriangleright_1 & \end{pmatrix}$	$\to$	$\begin{pmatrix} s_* & 0 & 0^k & s_+ \\ & \triangleright_1 & & \end{pmatrix}$
	Apply one step		Apply one step
$\to$	$\begin{pmatrix} s_+ & 0^k & 1 & s_* \\ & & \blacktriangleleft_2 & \end{pmatrix}$	$\to$	$\begin{pmatrix} s_* & 2 & 0^k & s_+ \\ & \blacktriangleright_2 & & \end{pmatrix}$
	Apply $M_{\blacktriangleleft}(k)$		Apply $M_{\blacktriangleright}(k)$
$\to^{f(k)}$	$\begin{pmatrix} s_+ & 0^k & 1 & s_* \\ \blacktriangleleft_1 & & & \end{pmatrix}$	$\to^{f(k)}$	$\begin{pmatrix} s_* & 2 & 0^k & s_+ \\ & & \blacktriangleright_1 & \end{pmatrix}$
	Apply one step		Apply one step
$\to$	$\begin{pmatrix} s_+ & 0^k & 1 & s_* \\ \triangleright_2 & & & \end{pmatrix}$	$\to$	$\begin{pmatrix} s_* & 2 & 0^k & s_+ \\ & & \triangleleft_2 & \end{pmatrix}$
	Apply $M_{\triangleright}(k)$		Apply $M_{\triangleleft}(k)$
$\to^{f(k)}$	$\begin{pmatrix} s_+ & 0^k & 1 & s_* \\ & \triangleright_1 & & \end{pmatrix}$	$\to^{f(k)}$	$\begin{pmatrix} s_* & 2 & 0^k & s_+ \\ \triangleleft_1 & & & \end{pmatrix}$
	Apply one step		Apply one step
$\to$	$\begin{pmatrix} s_+ & 0^k & 0 & s_* \\ & & \blacktriangleright_2 & \end{pmatrix}$	$\to$	$\begin{pmatrix} s_* & 0 & 0^k & s_+ \\ & \triangleright_2 & & \end{pmatrix}$
	Apply one step		Apply $M_{\triangleright}(k)$
$\to$	$\begin{pmatrix} s_+ & 0^k & 0 & s_* \\ & & & \blacktriangleright_1 \end{pmatrix}$	$\to^{f(k)}$	$\begin{pmatrix} s_* & 0 & 0^k & s_+ \\ & & & \triangleright_1 \end{pmatrix}$

□

3.1. *Action of SMART on cyclic tapes.* This section studies the action of SMART on cyclic tapes of length $\ell \geq 2$. We call *initial configurations* the following four cyclic configurations:

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

$$C_{\blacktriangleright} = \begin{bmatrix} 0 & 0^{\ell-1} \\ \blacktriangleright_1 & \end{bmatrix} \qquad C_{\blacktriangleleft} = \begin{bmatrix} 0 & 0^{\ell-1} \\ \blacktriangleleft_1 & \end{bmatrix}$$
$$C_{\triangleright} = \begin{bmatrix} 0 & 0^{\ell-1} \\ \triangleright_1 & \end{bmatrix} \qquad C_{\triangleleft} = \begin{bmatrix} 0 & 0^{\ell-1} \\ \triangleleft_1 & \end{bmatrix}$$

PROPOSITION 3.3. *Let $\ell \geq 2$. The action of the $(2 \cdot 3^{\ell})$th power of SMART on $C_{\blacktriangleright}$ and $C_{\triangleright}$ (respectively $C_{\blacktriangleleft}$ and $C_{\triangleleft}$) is a right-shift (respectively left-shift). Furthermore, the intermediate configurations are all distinct even up to a shift.*

Proof. By symmetries between $\blacktriangleright$ and $\blacktriangleleft$ (respectively $\triangleright$ and $\triangleleft$), we prove the result for $C_{\blacktriangleright}$ and $C_{\triangleright}$.

$$\begin{array}{ll}
 & \begin{bmatrix} 0 & 0 & 0^{\ell-2} \\ \blacktriangleright_1 & & \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0^{\ell-2} \\ \triangleright_1 & & \end{bmatrix} \\
 & \text{Apply one step} \qquad \text{Apply one step} \\
\to & \begin{bmatrix} 1 & 0 & 0^{\ell-2} \\ \blacktriangleleft_2 & & \end{bmatrix} \qquad \to \begin{bmatrix} 2 & 0 & 0^{\ell-2} \\ \blacktriangleright_2 & & \end{bmatrix} \\
 & \text{Apply } M_{\blacktriangleleft}(\ell-1) \qquad \text{Apply } M_{\blacktriangleright}(\ell-1) \\
\to^{f(\ell-1)} & \begin{bmatrix} 1 & 0 & 0^{\ell-2} \\ \blacktriangleleft_1 & & \end{bmatrix} \qquad \to^{f(\ell-1)} \begin{bmatrix} 2 & 0 & 0^{\ell-2} \\ \blacktriangleright_1 & & \end{bmatrix} \\
 & \text{Apply one step} \qquad \text{Apply one step} \\
\to & \begin{bmatrix} 1 & 0 & 0^{\ell-2} \\ \triangleright_2 & & \end{bmatrix} \qquad \to \begin{bmatrix} 2 & 0 & 0^{\ell-2} \\ \triangleleft_2 & & \end{bmatrix} \\
 & \text{Apply } M_{\triangleright}(\ell-1) \qquad \text{Apply } M_{\triangleleft}(\ell-1) \\
\to^{f(\ell-1)} & \begin{bmatrix} 1 & 0 & 0^{\ell-2} \\ \triangleright_1 & & \end{bmatrix} \qquad \to^{f(\ell-1)} \begin{bmatrix} 2 & 0 & 0^{\ell-2} \\ \triangleleft_1 & & \end{bmatrix} \\
 & \text{Apply one step} \qquad \text{Apply one step} \\
\to & \begin{bmatrix} 0 & 0 & 0^{\ell-2} \\ \blacktriangleright_2 & & \end{bmatrix} \qquad \to \begin{bmatrix} 0 & 0 & 0^{\ell-2} \\ \triangleright_2 & & \end{bmatrix} \\
 & \text{Apply one step} \qquad \text{Apply one step} \\
\to & \begin{bmatrix} 0 & 0 & 0^{\ell-2} \\ & \blacktriangleright_1 & \end{bmatrix} \qquad \to \begin{bmatrix} 0 & 0 & 0^{\ell-2} \\ & \triangleright_1 & \end{bmatrix}
\end{array}$$

We used moves $M_{\blacktriangleright}(\ell-1)$, $M_{\blacktriangleleft}(\ell-1)$, $M_{\triangleright}(\ell-1)$ and $M_{\triangleleft}(\ell-1)$ in patterns that overlap themselves on their first and last letters in the cyclic tape. This is valid, because the cell containing s_* is only visited at the last (respectively first) step of $M_{\blacktriangleright}$ and $M_{\blacktriangleleft}$ (respectively $M_{\triangleright}$ and $M_{\triangleleft}$).

For the last claim, by shift-commutation and bijectivity of the action, it is enough to show that a shifted copy of the initial configuration does not appear before the last step. This is clear from looking at the first columns, which have positive values on all but the first step and the two last steps. □

LEMMA 3.4. *For $\ell \geq 1$, the action of SMART on cyclic tapes of length ℓ is composed of four disjoint cycles of length $2\ell \cdot 3^{\ell}$, which are the orbits of the four initial configurations. Additionally, the action of the $(2 \cdot 3^{\ell})$th power of SMART on a cyclic tape is a right-shift (respectively left-shift) on the orbits of $C_{\blacktriangleright}$ and $C_{\triangleright}$ (respectively $C_{\blacktriangleleft}$ and $C_{\triangleleft}$).*

Proof. This is an immediate consequence of Proposition 3.3: the orbits are each of length $2\ell \cdot 3^{\ell}$ (number of shifts × number of steps for each shift), and are disjoint (by looking at

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

the first column in the previous proof). As there are $8\ell \cdot 3^\ell$ different cyclic configurations containing a head (eight different states with ℓ possible positions and a ternary tape of length ℓ), any configuration belongs to one of these four orbits: this concludes the proof. □

3.2. *Encoding SMART configurations.* Recall that for the SMART machine, $\Gamma = \{0, 1, 2\}$ and $Q \simeq \{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\} \times \{1, 2\}$. Lemma 3.4 implies that the set of cyclic SMART configurations of length ℓ that contain a head

$$\{x \in (\Gamma \cup (Q \times \Gamma))^{\mathbb{Z}/\ell\mathbb{Z}} \mid \text{ there exists } i \in \mathbb{Z}/\ell\mathbb{Z}, x_i \in Q \times \Gamma\}$$

is conjugate (as a finite dynamical system or as a permutation) to a disjoint union of four (depending on whether the head is in state $\blacktriangleright$, $\blacktriangleleft$, $\triangleright$, or $\triangleleft$) disjoint systems of counters ranging in $[\![0, \ell - 1]\!] \times \{1, 2\} \times \{0, 1, 2\}^\ell$ (respectively for the position of the head, the second component $\{1, 2\}$ of Q, and the tape of alphabet Γ). Each of these counters encodes $2\ell \cdot 3^\ell$ different values, which is exactly the length of any SMART cycle by Lemma 3.4.

We pick a natural conjugacy $E_\ell : C_\ell \to C_\ell$, a shift-invariant bijection that encodes a SMART configuration into its orbit position in base $2\ell \cdot 3^\ell$. We refer to the conjugacy E_ℓ as the *encoding map*.

More precisely, if $w \in C_\ell$ contains a head, then by Lemma 3.4, there exists some $0 \le n < 2\ell \cdot 3^\ell$ such that $w = (T_{\ell,\mathcal{S}})^n(C_q)$ for some initial configuration C_q ($q \in \{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\}$), and the map E_ℓ encodes the tuple (q, n) in C_ℓ as

$$E_\ell(w) = \sigma^{-\varepsilon \cdot a}\left(\begin{bmatrix} c_1 & c_2 & \dots & c_\ell \\ q_b & & & \end{bmatrix}\right)$$

where:

- q is stored in the first component of the state $\{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\}$;
- $b \cdot c \in \{1, 2\} \cdot \{0, 1, 2\}^\ell$ encodes $n \bmod 2 \cdot 3^\ell$, that is, c is a ternary word satisfying $v_{(3)}(c) = n \bmod 3^\ell$, and b stores $(\lfloor n/3^\ell \rfloor \bmod 2) + 1$ in the second component of the state;
- a is the quotient of n by $2 \cdot 3^\ell$, and is encoded in how much the cyclic configuration is shifted;
- $\varepsilon = +1$ if $q \in \{\blacktriangleright, \triangleright\}$ (respectively $\varepsilon = -1$ if $q \in \{\blacktriangleleft, \triangleleft\}$) shifts to the right (respectively left) if $q \in \{\blacktriangleright, \triangleright\}$ (respectively $q \in \{\blacktriangleleft, \triangleleft\}$),

and if $w \in C_\ell$ contains no head (that is, $w \in \Gamma^{\mathbb{Z}/\ell\mathbb{Z}}$), then we set $E_\ell(w) = w$.

In other words, given a configuration $w = (T_{\ell,\mathcal{S}})^n(C_q)$ for some initial configuration C_q ($q \in \{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\}$) and $0 \le n < 2\ell \cdot 3^\ell$, the map E_ℓ encodes the tuple (q, n) as plainly (and humanly readable) as possible.

In the next two sections (§§3.3 and 3.4), we detail how this bijection can be computed inductively, that is, we define piecewise-defined bijective maps F_{init}, $F_{k \to k+1}$ (for $0 \le k \le \ell - 2$), and $F_{\ell,\text{final}}$ acting on C_ℓ such that

$$E_\ell = F_{\ell,\text{final}} \circ \left(\prod_{k=0}^{\ell-2} F_{k \to k+1}\right) \circ F_{\text{init}}.$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

The precise definition of these maps is not strictly necessary to understand the rest of this paper (it is only used in Lemmas 4.18 and 4.20, where we need the piecewise defined functions to satisfy the requirements described in §4.2.3). On a first reading, we recommend the reader simply remembers the idea of encoding configurations into a counter, and goes directly to §4 about the finitary distortion of the SMART machine.

3.3. *Analysis of SMART configurations.* We now explain how, given a cyclic SMART configuration w of length ℓ, we determine which orbit it belongs in and its position in this orbit, that is, the number of steps required to obtain it from its corresponding initial configuration $C_{\blacktriangleright}$, $C_{\blacktriangleleft}$, $C_{\triangleright}$ or $C_{\triangleleft}$.

We say that a cyclic configuration $w \in C_\ell$ is performing the jth step of computation of $M_{\blacktriangleright}(k)$ (respectively $M_{\blacktriangleleft}(k)$, $M_{\triangleright}(k)$, $M_{\triangleleft}(k)$), for $0 \le k \le \ell - 1$ and $0 \le j \le f(k)$, if it contains the jth pattern of the sequence of transitions $M_{\blacktriangleright}(k)$ (respectively $M_{\blacktriangleleft}(k)$, $M_{\triangleright}(k)$, $M_{\triangleleft}(k)$) of Proposition 3.2. At this point, it may not be clear that this is unique, but this will follow from our argument.

If a configuration is performing some step of computation from one of the moves $M_{\blacktriangleright}(k)$, $M_{\blacktriangleleft}(k)$, $M_{\triangleright}(k)$, or $M_{\triangleleft}(k)$, we refer to this move as its *computation of level k*.

3.3.1. *Initialization.* We call the following patterns *special patterns of level k (for* $1 \le k \le \ell - 1$*)*:

$$s(\blacktriangleright_2, k) = \begin{pmatrix} s_+ & 0^{k-1} & 0 \\ & & \blacktriangleright_2 \end{pmatrix} \quad s(\blacktriangleright_1, k) = \begin{pmatrix} s_+ & 0^k & s_* \\ & & \blacktriangleright_1 \end{pmatrix}$$

$$s(\blacktriangleleft_2, k) = \begin{pmatrix} 0 & 0^{k-1} & s_+ \\ \blacktriangleleft_2 & & \end{pmatrix} \quad s(\blacktriangleleft_1, k) = \begin{pmatrix} s_* & 0^k & s_+ \\ \blacktriangleleft_1 & & \end{pmatrix}$$

$$s(\triangleright_2, k) = \begin{pmatrix} s_* & 0^k & s_+ \\ \triangleright_2 & & \end{pmatrix} \quad s(\triangleright_1, k) = \begin{pmatrix} 0 & 0^{k-1} & s_+ \\ \triangleright_1 & & \end{pmatrix}$$

$$s(\triangleleft_2, k) = \begin{pmatrix} s_+ & 0^k & s_* \\ & & \triangleleft_2 \end{pmatrix} \quad s(\triangleleft_1, k) = \begin{pmatrix} s_+ & 0^{k-1} & 0 \\ & & \triangleleft_1 \end{pmatrix}$$

and the following are *special patterns of level* ℓ:

$$\begin{pmatrix} 0 & 0^{\ell-1} \\ \blacktriangleright_2 & \end{pmatrix} \quad \begin{pmatrix} 0 & 0^{\ell-1} \\ \blacktriangleleft_2 & \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0^{\ell-1} \\ \triangleright_2 & \end{pmatrix} \quad \begin{pmatrix} 0 & 0^{\ell-1} \\ \triangleleft_2 & \end{pmatrix}$$

The latter appear exactly in the shifts of the configurations $\mathcal{S}^{-1}(C_q)$, for C_q the initial configurations ($q \in \{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\}$).

By the proof of Proposition 3.2, we see that if a cyclic configuration contains a special pattern $s(\blacktriangleright_2, k)$, $s(\blacktriangleright_1, k)$, $s(\blacktriangleleft_2, k)$, or $s(\blacktriangleleft_1, k)$ (respectively $s(\triangleright_2, k)$, $s(\triangleright_1, k)$, $s(\triangleleft_2, k)$, or $s(\triangleleft_1, k)$), then it performs the last two steps of $M_{\blacktriangleright}(k)$ or $M_{\blacktriangleleft}(k)$ respectively (respectively the first two steps of $M_{\triangleright}(k)$ or $M_{\triangleleft}(k)$).

CLAIM 3.5. *Given a cyclic configuration w of length ℓ containing a head, exactly one of the following holds:*

- *w is the shift of an initial configuration;*
- *w performs some step of computation of level 0 from either $M_{\blacktriangleright}(0)$, $M_{\blacktriangleleft}(0)$, $M_{\triangleright}(0)$, or $M_{\triangleleft}(0)$;*
- *w contains a special pattern of level $1 \le k \le \ell$.*

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

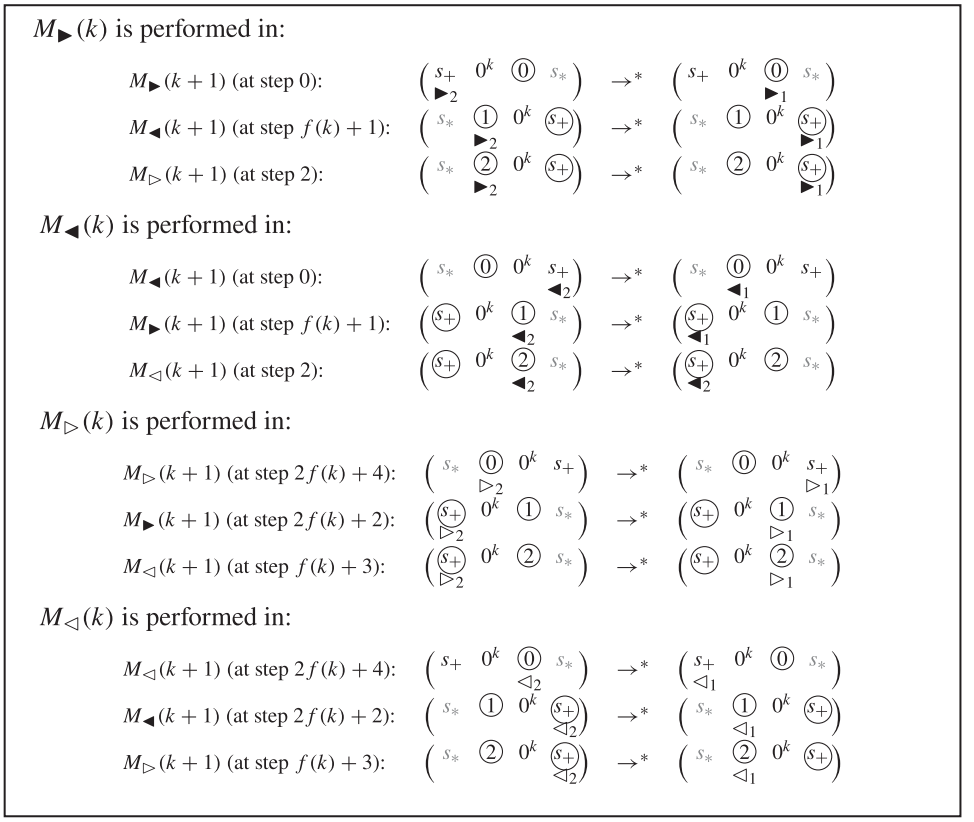


FIGURE 2. Bottom-up analysis of SMART configurations: $k \to k+1$. Intermediate steps of moves of level $k+1$ (patterns of length $k+3$) corresponding to sub-moves of level k. We darken the part of the pattern of length $k+2$ performing the sub-move of level k. Circled ternary letters are not modified by the sub-move of level k, and hence can be used to perform a case-analysis.

Proof. The patterns of $M_{\blacktriangleright}(0)$, $M_{\blacktriangleleft}(0)$, $M_{\triangleright}(0)$, and $M_{\triangleleft}(0)$ (eight in total), along with the special patterns of every level, disjointly cover all the non-initial configurations with a head (the level is determined by the distance to the nearest non-zero symbol in an appropriate direction). □

3.3.2. *Inductive analysis $k \to k+1$ (for $k+1 \leq \ell-1$).* Let $k \leq \ell-2$ be an integer and w be a non-initial cyclic configuration. If w performs some computation of level k (for $0 \leq k \leq \ell-2$), then it performs some computation of level $k+1$: indeed, Figure 2 shows that any computation of level k belongs to some computation of level $k+1$, and that the latter is uniquely determined by considering the value of two cells (circled on the figure) which are left unmodified by the computation of level k.

Additionally, if we know that w performs the jth step of some computation of level k (for $0 \leq j \leq f(k)$), then the same case-analysis determines j' ($0 \leq j' \leq f(k+1)$) such that w performs the j'th step of its computation of level $k+1$.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Example 3.6. For example, consider the cyclic configuration of length $\ell = 6$:

$$w = \left[\begin{array}{cccccc} 1 & 2 & 1 & 1 & 2 & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

First, w performs step 0 of $M_{\blacktriangleright}(0)$ (by simply considering all the computations of level 0). Indeed, the highlighted pattern inside the configuration below is the 0th pattern of the move $M_{\blacktriangleright}(0)$:

$$w = \left[\begin{array}{cccccc} 1 & 2 & 1 & \overbrace{1} & \overbrace{2} & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

Then, by looking at Figure 2 (four cases, with three subcases each), we deduce successively that:

(1) by considering Figure 2 (case $M_{\blacktriangleright}(0)$, second subcase),

$$w = \left[\begin{array}{cccccc} 1 & 2 & 1 & \textcircled{1} & \textcircled{2} & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

we deduce that w performs step $2 = 0 + (f(0) + 1)$ of $M_{\blacktriangleleft}(1)$. Additionally, the computation extends to the left, that is, the move $M_{\blacktriangleleft}(1)$ appears in the following highlighted pattern:

$$w = \left[\begin{array}{cccccc} 1 & 2 & \overbrace{1} & \overbrace{1} & \overbrace{2} & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

(2) by considering Figure 2 (case $M_{\blacktriangleleft}(1)$, third subcase),

$$w = \left[\begin{array}{cccccc} 1 & 2 & \textcircled{1} & 1 & \textcircled{2} & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

we deduce that w performs step $4 = 2 + 2$ of $M_{\triangleleft}(2)$. Additionally, the computation extends to the right and the move $M_{\triangleleft}(2)$ appears in the following highlighted pattern:

$$w = \left[\begin{array}{cccccc} 1 & 2 & \overbrace{1} & \overbrace{1} & \overbrace{2} & \overbrace{0} \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

(3) by considering Figure 2 (case $M_{\triangleleft}(2)$, first subcase),

$$w = \left[\begin{array}{cccccc} 1 & 2 & \textcircled{1} & 1 & 2 & \textcircled{0} \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

we deduce that w performs step $58 = 4 + (2f(2) + 4)$ of $M_{\triangleleft}(3)$. Additionally, the computation extends to the right and the move $M_{\triangleleft}(3)$ appears in the following highlighted pattern (remember that configurations are cyclic):

$$w = \left[\begin{array}{cccccc} \overbrace{1} & 2 & \overbrace{1} & \overbrace{1} & \overbrace{2} & \overbrace{0} \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

(4) by considering Figure 2 (case $M_{\triangleleft}(3)$, second subcase),

$$w = \left[\begin{array}{cccccc} \textcircled{1} & 2 & \textcircled{1} & 1 & 2 & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

we deduce that w performs step $218 = 58 + (2f(3) + 2)$ of $M_{\blacktriangleleft}(4)$. Additionally, the computation extends to the left and the move $M_{\blacktriangleleft}(4)$ appears in the following highlighted pattern:

$$w = \left[\begin{array}{cccccc} \overbrace{1} & \overbrace{2} & 1 & \overbrace{1} & 2 & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

(5) by considering Figure 2 (case $M_{\blacktriangleleft}(4)$, second subcase),

$$w = \left[\begin{array}{cccccc} \textcircled{1} & \textcircled{2} & 1 & 1 & 2 & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

we deduce that w performs step $460 = 218 + (f(4) + 1)$ of $M_{\blacktriangleright}(5)$.

On the final step, we should extend to the right; since we run out of cells on the tape, this means that we should interpret the circled 2-cell as both the first and last cell of the $M_{\blacktriangleright}(5)$-computation, that is, the move $M_{\blacktriangleright}(5)$ appears in the following highlighted self-overlapping pattern:

$$w = \left[\begin{array}{cccccc} \overbrace{1} & \overbrace{2} & 1 & \overbrace{1} & 2 & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

So we claim that w performs step $460 = 218 + (f(4) + 1)$ of $M_{\blacktriangleright}(5)$. And one can verify by a direct calculation (for example, by computer) that

$$\left(\begin{array}{cccccc} 2 & 0 & 0 & 0 & 0 & 0 \\ \blacktriangleright_2 & & & & & \end{array}\right)^{\infty} \to^{460} \left(\begin{array}{cccccc} 2 & 1 & 1 & 2 & 0 & 1 \\ & & \blacktriangleright_2 & & & \end{array}\right)^{\infty}$$

as we just deduced.

3.3.3. *Conclusion.* Let $w \in C_\ell$ be a cyclic tape of length ℓ containing a head. By the claim above, there are three different cases.

Either w is the shift of an initial configuration, in which case the state of w determines which orbit w belongs to, and counting the shift (and multiplying it by $2 \cdot 3^\ell$) is enough to know the position of the configuration w in its orbit. A similar reasoning applies for configurations w that contain a special pattern of level ℓ.

However, assume w contains some computation of level 0. Then w contains computations of every level k for $0 \le k \le \ell - 1$ by the previous induction. Similarly, if w contains a special pattern of level k for some $0 \le k \le \ell - 1$, then w corresponds to either the last two steps of $M_{\blacktriangleright}(k)$ or $M_{\blacktriangleleft}(k)$, or the first two steps of $M_{\triangleright}(k)$ or $M_{\triangleleft}(k)$. Then w corresponds to computations of every level k' for $k \le k' \le \ell - 1$ by the previous induction.

Finally, the structure of each SMART cycle (detailed in the proof of Proposition 3.3) enables to conclude about which orbit the configuration w belongs to, and its position in said orbit.

Example 3.7. Consider once again the cyclic configuration of length $\ell = 6$,

$$w = \left[\begin{array}{cccccc} 1 & 2 & 1 & 1 & 2 & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

By the previous example, w performs step 460 of $M_{\blacktriangleright}(5)$ on the following (self-overlapping) pattern:

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

$$w = \left[\begin{array}{cccccc} \overbrace{1} & \overbrace{2} & 1 & 1 & \overbrace{2} & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

Considering the proof of Proposition 3.3, only the orbits of $C_\triangleright$ and $C_\triangleleft$ have a cell containing tape-letter 2 that is left unchanged during a computation of level $\ell - 1$. Additionally, out of these two, only the orbit of $C_\triangleright$ contains the move $M_\blacktriangleright(5)$.

So w belongs in the orbit of $C_\triangleright$. From the proof of Proposition 3.3, we deduce that, modulo $2 \cdot 3^\ell$, the position of w in said orbit is $1 + 460$. Finally, 2 being the second cell of the tape, one shift already happened in the orbit: we conclude that the position of w in the orbit of $C_\triangleright$ is $2 \cdot 3^\ell + 461 = 1919$.

Indeed, one can verify by a direct calculation that

$$\left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \triangleright_1 & & & & & \end{array}\right] \to^{1919} \left[\begin{array}{cccccc} 1 & 2 & 1 & 1 & 2 & 0 \\ & & & \blacktriangleright_2 & & \end{array}\right]$$

as we just deduced.

3.4. *Encoding cyclic configurations into their orbit positions in C_ℓ.* Denote by $\mathcal{S}$ the SMART machine introduced above. Recall the encoding map $E_\ell : C_\ell \to C_\ell$ defined in §3.2 as

$$E_\ell(w) = \sigma^{-\varepsilon \cdot a}\left(\begin{bmatrix} c_1 & c_2 & \dots & c_\ell \\ q_b & & & \end{bmatrix}\right)$$

if w contains a head (in which case, there exists n and $q \in \{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\}$ such that $w = T^n_{\ell,\mathcal{S}}(C_q)$, and we set $b \cdot c \in \{1, 2\} \cdot \{0, 1, 2\}^\ell$ to encode $n \bmod 2 \cdot 3^\ell$, and a is the quotient of n by $2 \cdot 3^\ell$); and if $w \in C_\ell$ contains no head, then we set $E_\ell(w) = w$.

3.4.1. *Inductive encoding.* In this section, we use the analysis performed in §3.3 to provide a linear-time algorithm that computes this encoding $E_\ell : C_\ell \to C_\ell$ inductively.

Figure 3 describes a piecewise-defined bijection $F_{\text{init}} : C_\ell \to C_\ell$: each case describes how a pattern of length 2 (e.g. $\begin{pmatrix} 1 & s_* \\ \blacktriangleright_2 & \end{pmatrix}$) is bijectively replaced by another (in the previous example, by $\begin{pmatrix} 1 & s_* \\ & \blacktriangleright_1 \end{pmatrix}$). In other words, if a cyclic configuration w contains a sub-pattern of length 2 that matches with one case of Figure 3, then F_{init} replaces this sub-pattern in w by its image in the figure.

Similarly, Figures 4 and 5 together describe a piecewise-defined bijection $F_{k\to k+1} : C_\ell \to C_\ell$ that replaces sub-patterns of length $k + 4$. Intuitively, Figure 4 (defining the first half of $F_{k\to k+1}$) encodes moves of level $k + 1$ into counters, while Figure 5 (the second half of $F_{k\to k+1}$) encodes special patterns of level $k + 1$.

Finally, Figure 6 describes a similar bijection $F_{\ell,\text{final}} : C_\ell \to C_\ell$.

We can then prove the following lemma.

LEMMA 3.8. *Let w be a cyclic configuration of C_ℓ. Then,*

$$E_\ell(w) = \left[F_{\ell,\text{final}} \circ \prod_{k=0}^{\ell-2} F_{k\to k+1} \circ F_{\text{init}}\right](w).$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

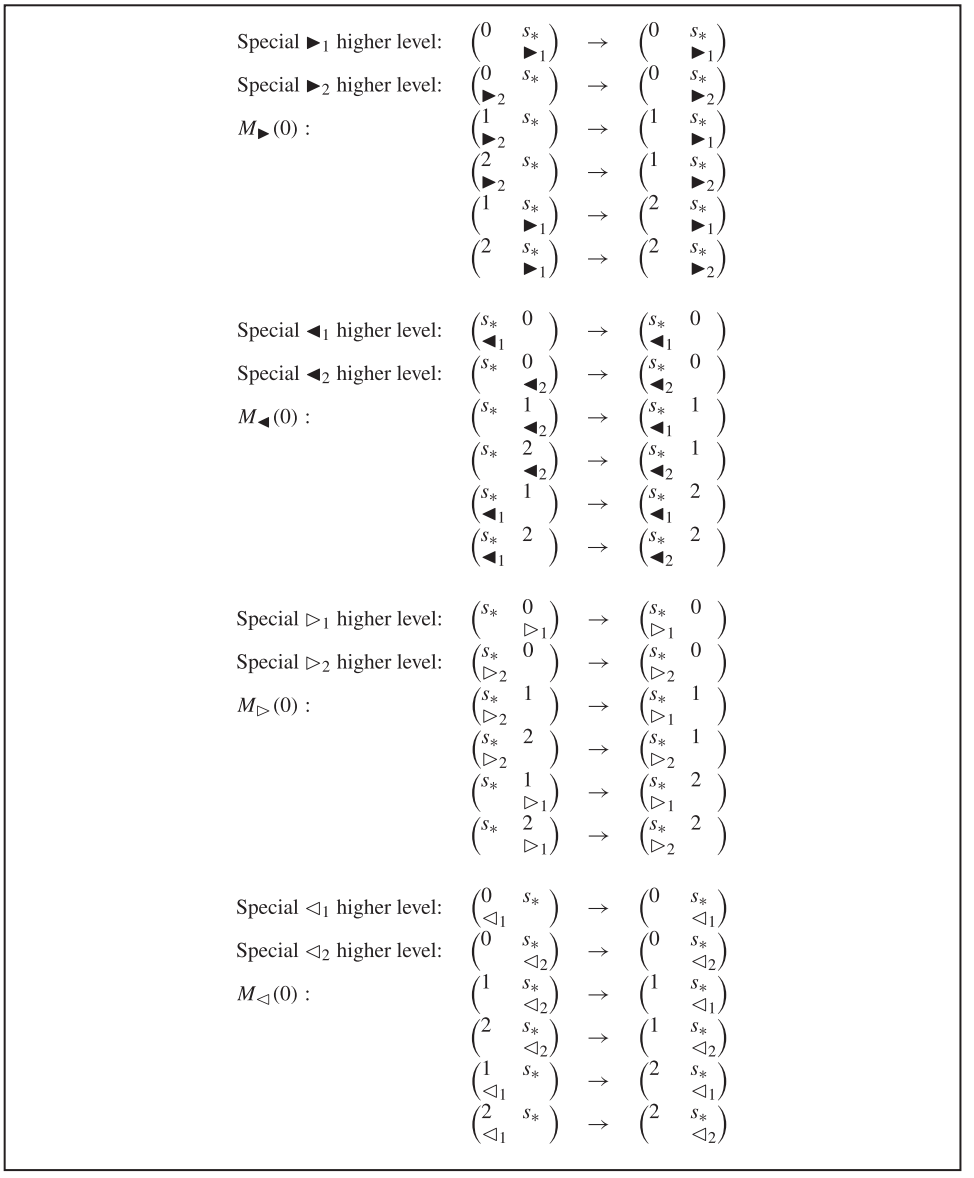


FIGURE 3. Encoding SMART configurations: F_{init}. F_{init} rewrites sub-patterns of length 2 in cyclic SMART configurations.

Sketch of proof. Let w be a configuration and k be some integer $k \leq \ell - 1$. If w is neither the shift of an initial configuration nor a special configuration of level $> k$, then there exists some $q \in \{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\}$ and a unique word $p = p_0 \cdots p_{k+1}$ of length $k + 2$ such that $p \sqsubseteq w$ and p computes the jth step of $M_q(k)$ for some $0 \leq j \leq f(k) = 3^{k+2} - 2$.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

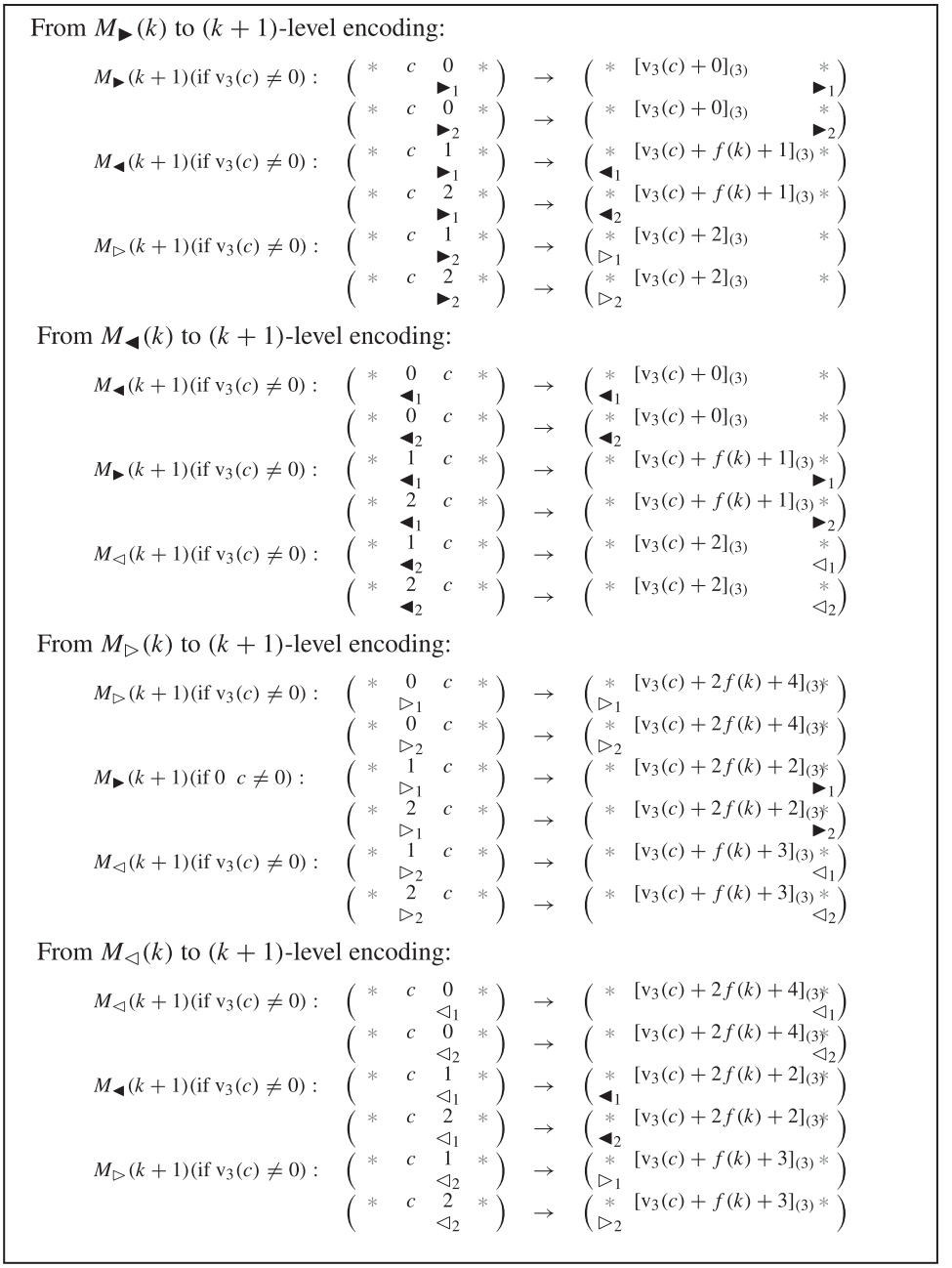


FIGURE 4. Encoding SMART configurations: $F_{k\to k+1}$ (Part 1: $k\to k+1$). First half of rewriting cases of $F_{k\to k+1}$, which rewrites sub-patterns of length $k+4$. Letters $*$ are unmodified. $F_{k\to k+1}$ extends the word $c\in\{0,1,2\}^{k+1}$ with one additional letter and, considering c as a counter, adds a number of steps in accordance with Lemma 3.2. Note that at $k+1=\ell-2$ (respectively $k+1=\ell-1$), the $*$-cells overlap on each other (respectively the counter), because we reach the length of the cyclic tape.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

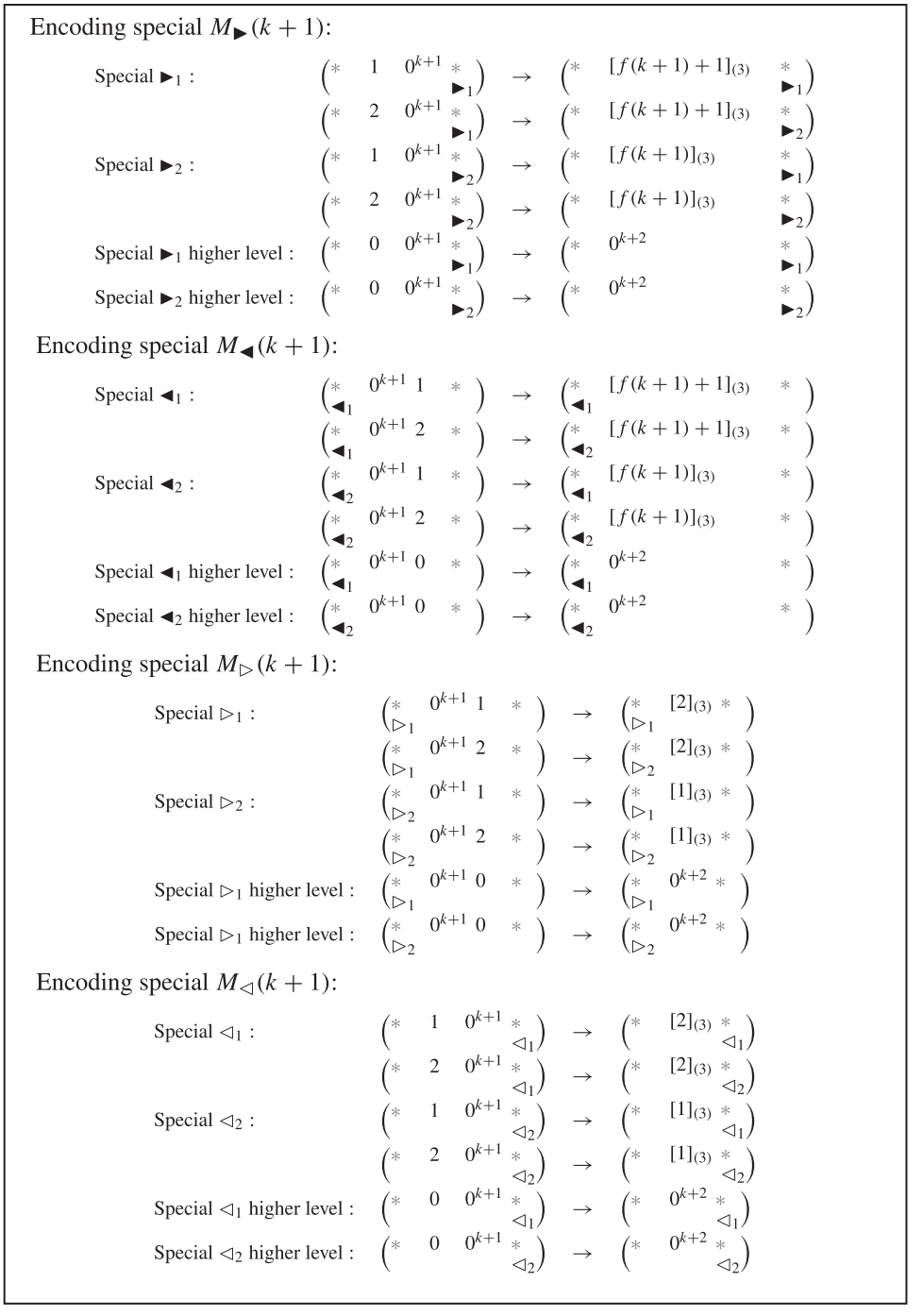


FIGURE 5. Encoding SMART configurations: $F_{k\to k+1}$ (Part 2: special $\to k+1$). Second half of rewriting cases of $F_{k\to k+1}$, which rewrites sub-patterns of length $k+4$. Letters $*$ are unmodified. Encodes special configurations of level $k+1$ by replacing the $k+2$ other letters by a counter of $\{0,1,2\}^{k+2}$ in accordance with Lemma 3.2, and preserves special configurations of level $> k+1$. Note that at $k+1=\ell-2$ (respectively $k+1=\ell-1$), the $*$-cells overlap on each other (respectively the counter), because we reach the length of the cyclic tape.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

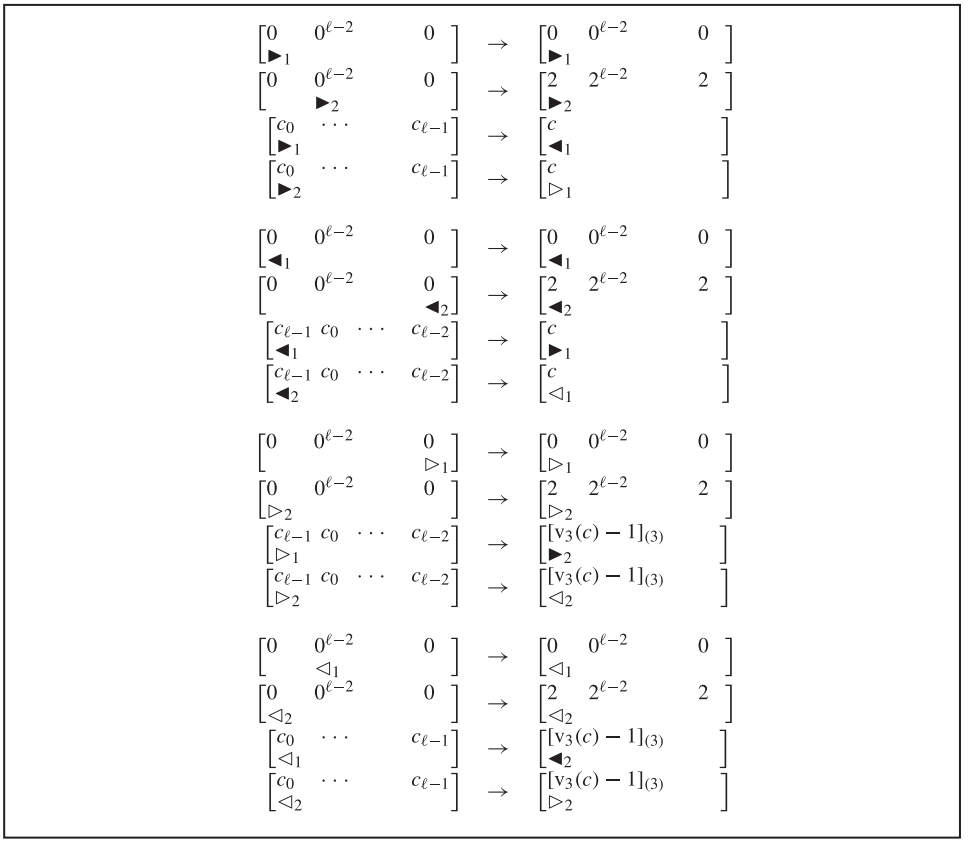


FIGURE 6. Final encoding step of SMART configurations: $F_{\ell,\mathrm{final}}$. Rewrites complete cyclic configurations of length ℓ. $F_{\ell,\mathrm{final}}$ acts according to the proof of Proposition 3.3: it maps encodings of level $\ell-1$ to their final encodings, and 'corrects' the position the head and shifts the counter when required (in the encodings of $M_{\blacktriangleleft}(\ell-1)$ and $M_{\triangleright}(\ell-1)$, or in initial configurations, whose heads were moved when applying F_{init}).

Then by induction on k, one sees that the partial composition

$$\prod_{k'=0}^{k-1} F_{k'\to k'+1} \circ F_{\mathrm{init}},$$

when applied on w, replaces p in w by another pattern p' of the same length defined as follows:

$$p' = \begin{cases} \begin{pmatrix} c_0 & \cdots & c_k & p_{k+1} \\ & & & \blacktriangleright_b \end{pmatrix} & (\text{where } b = p_0 \in \{1,2\}) \text{ if } p \text{ performs } M_{\blacktriangleright}(k), \\ \begin{pmatrix} p_0 & c_0 & \cdots & c_k \\ \blacktriangleleft_b & & & \end{pmatrix} & (\text{where } b = p_{k+1} \in \{1,2\}) \text{ if } p \text{ performs } M_{\blacktriangleleft}(k), \\ \begin{pmatrix} p_0 & c_0 & \cdots & c_k \\ \triangleright_b & & & \end{pmatrix} & (\text{where } b = p_{k+1} \in \{1,2\}) \text{ if } p \text{ performs } M_{\triangleright}(k), \\ \begin{pmatrix} c_0 & \cdots & c_k & p_{k+1} \\ & & & \triangleleft_b \end{pmatrix} & (\text{where } b = p_0 \in \{1,2\}) \text{ if } p \text{ performs } M_{\triangleleft}(k), \end{cases}$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

where $c \in \{0, 1, 2\}^{k+1}$ is a ternary counter such that $\mathrm{v}_3(c) = j + 1$. Notice that $0 \leq j \leq f(k) - 2$, where $f(k) = 3^{k+2} - 2$; so that $1 \leq j + 1 \leq f(k) - 1$ fits exactly in the space of non-zero counters. The zero counters are reserved for (shifts of) initial and special configurations.

Finally,

$$F_{\ell,\mathrm{final}} \circ \prod_{k=0}^{\ell-2} F_{k\to k+1} \circ F_{\mathrm{init}}$$

is equal to E_ℓ by considering how $F_{\ell,\mathrm{final}}$ acts in accordance with the structure of the four disjoint cycles of SMART (detailed in the proof of Proposition 3.3). □

Example 3.9. Consider once again the cyclic configuration of length $\ell = 6$ from Example 3.6, and let us use the formulas above to encode it into a counter value. This process will roughly mirror Examples 3.6 and 3.7, except that due to our coding convention, the counter value is one larger than the correct one until the very last step. First, we apply F_{init}, which observes based on the highlighted cells

$$w = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 0 \\ & & & \blacktriangleright_2 & & \end{bmatrix}$$

that the first case of $M_\blacktriangleright(0)$ applies, and rewrites this as

$$w = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 0 \\ & & & & \blacktriangleright_1 & \end{bmatrix}$$

Next, we apply $F_{0\to 1}$. Based on the highlighted cells, the fourth case of $M_\blacktriangleright(0)$ applies and we rewrite this as

$$w = \begin{bmatrix} 1 & 2 & 1 & \overbrace{1 & 0}^{f(0)} & 0 \\ & & \blacktriangleleft_2 & & & \end{bmatrix} \quad \mathrm{v}_3(1) + f(0) + 1 = 3$$

Next, we apply $F_{1\to 2}$. Based on the highlighted cells, the fifth case of $M_\blacktriangleleft(1)$ applies and we rewrite this as

$$w = \begin{bmatrix} 1 & 2 & \overbrace{0 & 1 & 2}^{\mathrm{v}_3(10)} & 0 \\ & & & & \triangleleft_1 \end{bmatrix} \quad \mathrm{v}_3(10) + 2 = 5$$

Next, we apply $F_{2\to 3}$. Based on the highlighted cells, the first case of $M_\triangleleft(2)$ applies and we rewrite this as

$$w = \begin{bmatrix} 1 & 2 & \overbrace{2 & 0 & 1 & 2}^{f(2)} \\ \triangleleft_1 & & & & & \end{bmatrix} \quad \mathrm{v}_3(012) + 2f(2) + 5 = 59$$

Next, we apply $F_{3\to 4}$. Based on the highlighted cells, the third case of $M_\triangleleft(3)$ applies and we rewrite this as

$$w = \begin{bmatrix} \overbrace{0}^{\mathrm{v}_3(2012)} & 2 & \overbrace{2 & 2 & 0 & 1}^{f(3)} \\ & \blacktriangleleft_1 & & & & \end{bmatrix} \quad \mathrm{v}_3(2012) + 2f(3) + 2 = 219$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Next, we apply $F_{4\to 5}$. Based on the highlighted cells, the fourth case of $M_{\blacktriangleleft}(4)$ applies and we rewrite this as

$$w = \left[\begin{array}{cccccc} \overbrace{2}^{\mathrm{v}_3(22010)+f(4)+1=461} & \overbrace{1}^{} & 2 & 2 & 0 & 0 \\ & \blacktriangleright_2 & & & & \end{array}\right]$$

Finally, we apply F_{final} (in Figure 6). We are in the fourth case, where we do not touch the counter and just change the state to $\triangleright_1$:

$$w = \left[\begin{array}{cccccc} 2 & 1 & 2 & 2 & 0 & 0 \\ & \triangleright_1 & & & & \end{array}\right]$$

This gives the expected result: as $v_3(122002) = 461$, and the head is shifted once to the right in the tape, this configuration has position $1 \cdot (2 \cdot 3^{\ell}) + 461 = 1919$ in the orbit of the initial configuration corresponding to state $\triangleright_1$.

4. *Finitary distortion for SMART*

In this section, we first introduce the group $\mathcal{G}_\ell$ generated by Turing machines instructions (§4.1.2), and we slightly alter the SMART machine $\mathcal{S}$ (§4.1.1): we call it the *decorated SMART* since we add some additional components to its states. The action of the SMART machine extends trivially to the new decorations—new components are added as a Cartesian product, and the head simply carries its new components without modifying or reading them.

Denoting by $T_{\ell,\mathcal{S}} : C_\ell \to C_\ell$ the finite action of this decorated SMART on the cyclic tapes of C_ℓ, we then establish Lemma 4.2, an intermediary result about $T_{\ell,\mathcal{S}}$. Namely, this automorphism is 'finitarily distorted' in $\mathcal{G}_\ell$, in the sense that all its powers (including those exponential in ℓ) have word norm polynomial in ℓ under the fixed generators (which is exponentially lower than the order of the group would suggest). The rest of this section is dedicated to the proof of this lemma.

Later in §5, we prove that every non-trivial full shift contains a distortion element of infinite order. Technically, §5 focuses on transporting the finite actions $T_{\ell,\mathcal{S}}$ of the decorated SMART into an infinite action on a full shift (and showing that this transposition preserves distortion). In other words, the 'distortion' aspect of Lemma 5.1 entirely comes from Lemma 4.2, that is, from the main result of this section.

4.1. *Context and results*

4.1.1. *Decorated SMART on C_ℓ.* Let $\mathcal{S}^{(\mathrm{o})} = (Q^{(\mathrm{o})}, \Gamma^{(\mathrm{o})}, \Delta^{(\mathrm{o})})$ be the SMART machine (see §3). Define the *decorated version* of the SMART machine as $\mathcal{S}_{\text{dec}} = (Q, \Gamma, \Delta)$, with

$$\begin{aligned} Q &= Q^{(\mathrm{o})} \times D \times G, \\ \Gamma &= \Gamma^{(\mathrm{o})}, \\ \Delta &= \bigcup_{(d,x)\in D\times G} \{((q,d,x),a,(q',d,x),b) : (q,a,q',b) \in \Delta^{(\mathrm{o})}\} \\ &\quad \cup \bigcup_{(d,x)\in D\times G} \{((q,d,x),\delta,(q',d,x)) : (q,\delta,q') \in \Delta^{(\mathrm{o})}\} \end{aligned}$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

for $D = \{d_1, d_2\}$ and $G = [\![0, 5]\!]$, that is, the states of SMART now carry a state $q^{(\mathrm{o})} \in Q^{(\mathrm{o})}$ of the original machine $\mathcal{S}$, a special symbol $d \in D$ called the *duck*, and a *ghost symbol* $x \in G$. We have $|Q| = 96$.

Technically, the two SMART machines $\mathcal{S}_{\mathrm{dec}}$ and $\mathcal{S}$ are different: for one, they act on different sets of cyclic tapes (since they have different sets of states). However, they have very similar behaviors, as the decorated machine only carries its decoration unmodified in its state while acting on tapes. To refer to the original set of states of SMART, we will use $Q^{(\mathrm{o})}$, and Q will denote $Q = Q^{(\mathrm{o})} \times D \times G$.

Since the remainder of this article only uses the decorated version of SMART, in what follows, $\mathcal{S}$ will refer to the decorated version of the machine, despite lacking the 'dec' subscript. This should not cause confusion, as the machines act on different sets.

The point of the *ghost* $[\![0, 5]\!]$ is to allow us to condition the application of gates, and to build the permutations we perform in §4.2. The *duck* $d \in \{d_1, d_2\}$ will be important during intermediate steps of computation in §4.2, to realize piecewise defined functions.

For $S \subseteq C_\ell$ a subset of finite cyclic tapes, we denote by $S[d_1]$ and $S[d_2]$ the subsets of the tapes of S containing a head, and whose ducks respectively are $d = d_1$ and $d = d_2$. For $d \in \{d_1, d_2\}$ and a function $f : S \to S$, we abuse notation and denote by $f\!/_d$ the extended restriction $f\!/_{S[d]}$ of f to $S[d]$:

$$f\!/_d(w) = \begin{cases} f(w) & \text{if } w \in S[d], \\ w & \text{otherwise.} \end{cases}$$

4.1.2. *Group of Turing machine instructions on finite cyclic tapes.* Recall that C_ℓ is the set of finite cyclic tapes of length $\ell \geq 2$ (see §2) with states Q and tape-alphabet Γ, containing at most one head (that is, a letter in $Q \times \Gamma$).

Let $\mathcal{G}_\ell$ be the finitely generated subgroup of $\mathrm{Sym}(C_\ell)$ generated by state-dependent moves, and the unary gates permuting heads. Formally, for $g \in \mathrm{Sym}(Q \times \Gamma)$, define the *unary gate* $\pi_g \in \mathrm{Sym}(C_\ell)$ as

$$\pi_g(w)_j = \begin{cases} g(w_j) & \text{if } w_j \in (Q \times \Gamma), \\ w_j & \text{if } w_j \in \Gamma, \end{cases}$$

for a cyclic tape $w \in C_\ell$. In addition, for $q \in Q$, define the *state-dependent right move* $\rho_q \in \mathrm{Sym}(C_\ell)$ as

$$\rho_q(w)_j = \begin{cases} (q, w_j) & \text{if } w_{j-1 \bmod \ell} \in (\{q\} \times \Gamma), \\ \pi_\Gamma(w_j) & \text{if } w_{j-1 \bmod \ell} \notin (\{q\} \times \Gamma) \wedge w_j \in (\{q\} \times \Gamma) \cup \Gamma, \\ w_j & \text{if } w_{j-1 \bmod \ell} \notin (\{q\} \times \Gamma) \wedge w_j \in ((Q \setminus \{q\}) \times \Gamma), \end{cases}$$

for $w \in C_\ell$ and $\pi_\Gamma : \Gamma \cup (Q \times \Gamma) \to \Gamma$ the natural projection.

We then define the group $\mathcal{G}_\ell \leq \mathrm{Sym}(C_\ell)$ generated by these permutations:

$$\mathcal{G}_\ell = \langle \{\pi_g : g \in \mathrm{Sym}(Q \times \Gamma)\} \cup \{\rho_q \mid q \in Q\} \rangle.$$

We can see the group $\mathcal{G}_\ell$ as the group generated by the instructions of Turing machines: moving heads based on their states, or permuting their values. To ease notation, we denote

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

$\prod_{q\in Q}\rho_q$ by ρ. This (finite) group is equipped with a metric, that is, the word norm given by the generators π_g and ρ_q.

It is easy to see that for any reversible Turing machine $\mathcal{M}$ of states Q and tape-alphabet Γ, $T_{\ell,\mathcal{M}}$ is an element of $\mathcal{G}_\ell$. Indeed, a step of computation is the composition of a head permutation α of $Q\times\Gamma$, followed with state-dependent moves β_{+1} and β_{-1}:

$$\alpha(q,a)=\begin{cases}(q',b) & \text{if } (q,a,q',b)\in\Delta,\\ (q',a) & \text{if } (q,\pm 1,q')\in\Delta,\end{cases}$$

$$\beta_{+1}=\prod_{q'\mid \text{there exists } q,(q,+1,q')\in\Delta}\rho_{q'},$$

$$\beta_{-1}=\prod_{q'\mid \text{there exists } q,(q,-1,q')\in\Delta}{\rho_{q'}}^{-1}.$$

Finally, we denote by $\delta(\ell,n)$ the word norm of $(T_{\ell,\mathcal{M}})^n$ in $\mathcal{G}_\ell$. In this section, we focus on proving that $\delta(\ell,n)$ is polynomial in ℓ for powers of the decorated SMART machine (even for powers exponential in ℓ).

Remark 4.1. It can be shown that $\mathcal{G}_\ell$ is, for large enough ℓ, $|Q|$, and $|\Gamma|$ (in particular for all versions of the SMART machine we consider and for $\ell\geq 2$), of bounded index in the automorphism group of C_ℓ under the shift action of $\mathbb{Z}_\ell$. This is not particularly useful, however, as what we need it for is to provide a group where the SMART machine corresponds to an element of small word norm (far smaller than the radius of the group).

4.1.3. *Main result: finitary distortion of the decorated SMART.* Recall that $m:\mathbb{N}\to\mathbb{N}$ is the *movement function*, that is, $m(n)$ is the maximal number of cells the machine $\mathcal{S}$ can visit in n steps; and that $\delta(\ell,n)$ is the word norm of $(T_{\ell,\mathcal{S}})^n$ in $\mathcal{G}_\ell$.

LEMMA 4.2. *Let $\mathcal{S}$ be the (decorated) SMART machine.*

(1) *$T_{\mathcal{S}}$ has infinite order.*
(2) *There exist some $C, C'>0$ such that $m(n)\leq C\log n+C'$.*
(3) *There exists some $p>0$ such that $\delta(\ell,n)=O(\ell^p)$.*

In fact, for $\mathcal{S}$, we can take $C=\ln(2)/\ln(3)$ and $p=4$.

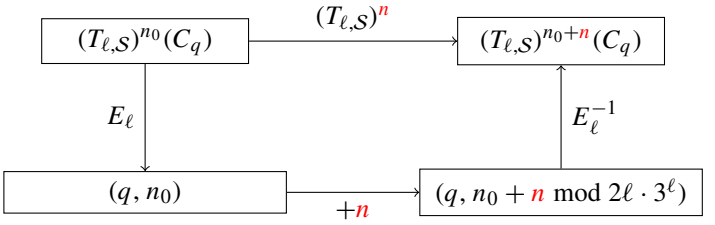
Any finite order T satisfies the latter two items, and any non-trivial state-dependent shift satisfies the first and the third items. Achieving the first two items is already difficult and to our knowledge, these properties have only been explicitly shown (in the reversible case) for the SMART machine and the binary SMART machine [15]. We expect that the Kari–Ollinger construction in [32] can be used to produce more examples of machines satisfying these two properties (at least $m(n)=O(n/\log n)$ follows from general principles for all these machines [28]).

4.1.4. *Proof of Lemma 4.2.* For the second item, the logarithmic speed of SMART is well known. To sketch a proof, consider the following computation: after less than 18 steps, the head of SMART is in state $\triangleright_1$ or $\triangleleft_1$ reading a 0 (ignoring the ghost and the duck). Then SMART is either at the left (for $\triangleright_1$) or right (for $\triangleleft_1$) extremity of some word

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

0^m for some $m \geq \log_3(k) + 2$ or by [14, Lemma 4], it builds around this position some pattern in the set C_m for $m \geq \log_3(k) + 2$ (with the notation of [14, Lemma 4]). Either way, after this point, k steps of computations cannot read more than $\log_3(k) + 2$ different cells.

The proof of the third item is a matter of programming powers of the machine efficiently with the generators of $\mathcal{G}_\ell$, which can be considered as the primitive reversible instructions of Turing machines. To achieve this, we encode configurations into their orbit position with the automorphism E_ℓ (defined in §3.2), perform an addition on these positions (as defined below), and decode back, as summarized in the commuting diagram below:



More precisely, we prove the two following lemmas.

LEMMA 4.3. *Let $E_\ell : C_\ell \to C_\ell$ be the encoding map defined in §3.2. There exists some $\tilde{E}_\ell : C_\ell \to C_\ell$ in $\mathcal{G}_\ell$ with word norm $O(\ell^4)$ such that*

$$w \in C_\ell[d_1] \implies \tilde{E}_\ell(w) = E_\ell(w),$$

where $C_\ell[d_1]$ is the set of cyclic tapes $w \in C_\ell$ having a head with duck $d = d_1$.

Note that we say nothing about the action of $\tilde{E}_\ell$ on tapes having duck $d = d_2$. This restriction comes from our use of the *ducking trick* to build $\tilde{E}_\ell$, which produces 'garbage' (that is, acts with no reasonable interpretation) on heads having duck $d = d_2$. See §4.2.3 for more details.

LEMMA 4.4. *Let $(+n)_\ell$ be the bijection of C_ℓ that performs the addition of $n \in \mathbb{N}$ in base $2\ell \cdot 3^\ell$ on the orbits positions encoded by E_ℓ. Recall that $(+n/_{d_1})_\ell : C_\ell \to C_\ell$ is defined as*

$$(+n/_{d_1})_\ell(w) = \begin{cases} (+n)_\ell(w) & \text{if } w \in C_\ell[d_1], \\ w & \text{otherwise.} \end{cases}$$

Then $(+n/_{d_1})_\ell$ belongs to $\mathcal{G}_\ell$ with word norm $O(\ell^3)$.

We give a more detailed definition of $(+n)_\ell$ in §4.3.2. Informally, $(+n)_\ell$ adds n to the counter $E_\ell(w)$ that encodes the orbit position of the SMART configuration w.

Then, these two lemmas are enough to prove the third item of Lemma 4.2 about the decorated SMART being distorted on cyclic tapes of length ℓ (more precisely, $\delta(\ell, n) = O(\ell^4)$). Indeed, combining the two previous results, we obtain the following lemma.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

LEMMA 4.5. *Denoting again:*

$$T_{\ell,\mathcal{S}/d_1}(w) = \begin{cases} T_{\ell,\mathcal{S}}(w) & \text{if } w \in C_\ell[d_1], \\ w & \text{otherwise,} \end{cases}$$

then for any $n \in \mathbb{N}$, $(T_{\ell,\mathcal{S}/d_1})^n$ *belongs to* $\mathcal{G}_\ell$ *with word norm* $O(\ell^4)$.

Proof. Let $(+n/_{d_1})_\ell$ be given by Lemma 4.4. With Lemma 4.3, one can conjugate $(+n/_{d_1})_\ell$ with $\tilde{E}_\ell$ and obtain a bijection on C_ℓ that maps configurations of $C_\ell[d_1]$ to their nth iterate by $\mathcal{S}$ and is the identity on $C_\ell[d_2]$. In other words,

$$(T_{\ell,\mathcal{S}/d_1})^n = ((+n/_{d_1})_\ell)^{\tilde{E}_\ell}.$$

Indeed, the addition of $(+n/_{d_1})_\ell$ is only performed on heads having duck d_1, so the garbage generated by $\tilde{E}_\ell$ on ducks $d = d_2$ is canceled in the conjugation; additionally, the shift of the tape (which happens when the addition modulo $2 \cdot 3^\ell$ overflows) is performed to the right (respectively to the left), exactly like $(T_{\ell,\mathcal{S}})^{2\cdot 3^\ell}$ acts on configurations $C_\blacktriangleright$ and $C_\triangleright$ (respectively $C_\blacktriangleleft$ and $C_\triangleleft$). □

And this lemma then leads to the following proof.

Proof of Lemma 4.2, third item. Let $d' = d_1 \leftrightarrow d_2 \in \mathrm{Sym}(D)$ be the involution that swaps ducks d_1 and d_2, and $d = \mathrm{id} \times d' \times \mathrm{id} \times \mathrm{id} \in \mathrm{Sym}(H)$ its lift to $H = Q \times \Gamma$ (where $Q = Q^{(\mathrm{o})} \times D \times G$). We have

$$(T_{\ell,\mathcal{S}})^n = (\pi_d \circ (T_{\ell,\mathcal{S}/d_1})^n \circ \pi_d) \circ (T_{\ell,\mathcal{S}/d_1})^n$$

because $(T_{\ell,\mathcal{S}/d_1})^{\pi_d} = T_{\ell,\mathcal{S}/d_2}$. □

4.1.5. *Overview.* The following subsections deal with the proofs of Lemmas 4.3 and 4.4, which respectively prove that the encoding and the addition can be implemented in $\mathcal{G}_\ell$ with respective word norms $O(\ell^4)$ and $O(\ell^3)$.

More precisely, §4.2 contains our main technical results, which define and implement conditional permutations. It also contains an exposition of the *ducking trick*, a method we use to implement piecewise-defined bijections. Section 4.3 contains the proofs of Lemmas 4.3 and 4.4.

4.2. *Permutation engineering in* C_ℓ. In this section, we develop two methods. In §4.2.1, we consider *permutation conditioning*, which consists in building permutations $\pi_{g,C}$ (for some $g \in \mathrm{Sym}(Q \times \Gamma)$ a permutation of the head, and some $C \subseteq (Q \times \Gamma) \times \Gamma^{\ell-1}$) that apply the permutation g if the condition C holds. We prove (Lemma 4.10) that if C has a simple enough description, then $\pi_{g,C}$ has polynomial word norm in ℓ. In §4.2.3, we consider the *ducking trick*, which allows us to efficiently build piecewise-defined bijections.

4.2.1. *Permutation conditioning.* We call *conditions* the subsets C of $(Q \times \Gamma) \times \Gamma^{\ell-1}$. For $g \in \mathrm{Sym}(Q \times \Gamma)$, if C (considered as a subset of C_ℓ) is a π_g-invariant subset, we can

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

define the bijection $\pi_{g,C} : C_\ell \to C_\ell$ as $\pi_{g,C}(w) = w$ if $w \in C_\ell$ contains no head; and if $w \in C_\ell$ contains a head at position, say, $i_0 \in \mathbb{Z}/\ell\mathbb{Z}$, we set $\pi_{g,C}(w)$ to

$$\pi_{g,C}(w)_i = \begin{cases} w_i & \text{if } i \neq i_0, \\ g(w_i) & \text{if } i = i_0 \text{ and } w_i \cdot w_{i+1} \cdots w_{\ell-1} \cdot w_0 \cdots w_{i-1} \in C, \\ w_i & \text{if } i = i_0 \text{ and } w_i \cdot w_{i+1} \cdots w_{\ell-1} \cdot w_0 \cdots w_{i-1} \notin C. \end{cases}$$

We then call $\pi_{g,C}$ the *conditional application of g under condition C*.

Recall that the states of Q have a ghost component G. We split Q into two components: $Q = Q' \times G$ (so, with the notation $Q = Q^{(o)} \times D \times G$, we have $Q' = Q^{(o)} \times D$; but the exact structure of Q' has no importance in this section).

Then, the set $H = Q \times \Gamma$ also splits into $H = H' \times G$, where $H' = Q' \times \Gamma$. Note that for any permutation $g \in \mathrm{Sym}(G)$, the permutation $\mathrm{id}_{H'} \times g$ belongs to $\mathrm{Sym}(H)$; and similarly, if $g \in \mathrm{Sym}(H')$, the permutation $g \times \mathrm{id}_G$ belongs to $\mathrm{Sym}(H)$. Finally, all conditions we define below will be of the form $C = C' \times G$, for $C' \subseteq H' \times \Gamma^{\ell-1}$.

In this section, we prove Lemma 4.7: for gates $g \in \mathrm{Sym}(H')$ and $\pi_{g\times\mathrm{id}}$-invariant conditions $C = C' \times G$ (for some $C' \subseteq H' \times \Gamma^{\ell-1}$), the conditioned gates $\pi_{g\times\mathrm{id},C}$ belong to $\mathcal{G}_\ell$. We also provide upper bounds on the word norm of $\pi_{g\times\mathrm{id},C}$ depending on C. This is essentially Barrington's theorem [4].

As a first step, we consider the opposite case: instead of leaving the ghost component of the head intact while permuting, we consider permutations that only permute the ghost component G. As conditions $C = C' \times G$ (for $C' \subseteq H' \times \Gamma^{\ell-1}$) are trivially $\pi_{\mathrm{id}\times g}$-invariant for any $g \in \mathrm{Sym}(G)$, the conditioned gates $\pi_{\mathrm{id}\times g,C}$ are always defined and we obtain the following lemma.

LEMMA 4.6. *For any $g \in \mathrm{Alt}(G)$ and condition $C = C' \times G$ (for some $C' \subseteq H' \times \Gamma^{\ell-1}$), the conditioned gate $\pi_{\mathrm{id}\times g,C}$ belongs to $\mathcal{G}_\ell$. Let $T : \mathcal{P}(H \times \Gamma^{\ell-1}) \to \mathbb{N}$ be the optimal function such that $\|\pi_{\mathrm{id}\times g,C}\| \leq T(C)$ for all $g \in \mathrm{Alt}(G)$. Then T satisfies the following inequalities:*

$$\begin{aligned} T([a]_j \times G) &\leq |\min(j, \ell - j)|, \\ T((C' \cap C'') \times G) &\leq 2(T(C' \times G) + T(C'' \times G)), \\ T((C' \cup C'') \times G) &\leq \begin{cases} T(C' \times G) + T(C'' \times G) & \text{if } C' \cap C'' = \emptyset, \\ 2(T(C' \times G) + T(C'' \times G)) + 5 & \text{otherwise}, \end{cases} \\ T((C'^c) \times G) &\leq T(C' \times G) + 1. \end{aligned}$$

Proof. We prove by induction over $C' \subseteq H' \times \Gamma^{\ell-1}$ that, denoting $C = C' \times G$, every $\pi_{\mathrm{id}\times g,C}$ (for $g \in \mathrm{Alt}(G)$) has word norm that checks the aforementioned inequalities.

Case 1. If $C' = B' \times \Gamma^{\ell-1}$ for some $B' \subseteq H'$, then any such $\pi_{\mathrm{id}\times g,B'\times G}$ already appears in the set of generators of $\mathcal{G}_\ell$.

Case 2. If $C' = [a]_j \triangleq H' \times \Gamma^{j-1} \times \{a\} \times \Gamma^{\ell-j-1}$ for some $j \in [\![1, \ell]\!]$ and $a \in \Gamma$, define $B' = (Q' \times \{a\}) \times \Gamma^{\ell-1}$. Then one can conjugate $\pi_{\mathrm{id}\times g,B'\times G}$ (which belongs to $\mathcal{G}_\ell$ by the first item) with either the right-move ρ^j or the left-move $\rho^{-(\ell-j)}$: the resulting

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

permutation applies g on the ghost symbol if and only if j cells away from the head, the content of the tape is a.

Case 3. If $C' = (C'_1)^c$, then $\pi_{\mathrm{id}\times g, C'\times G} = \pi_{\mathrm{id}\times g^{-1}, C'_1\times G} \circ \pi_{\mathrm{id}\times g}$.

Case 4. If $C' = C'_1 \cap C'_2$, we use the 'commutator trick': as G has cardinality at least 5, g is a commutator by Ore's theorem [42, Theorem 7], so there exist g_1, g_2 such that $g = [g_1, g_2]$. By the induction hypothesis and a straightforward calculation, we conclude that

$$\pi_{\mathrm{id}\times g, (C'_1\cap C'_2)\times G} = [\pi_{\mathrm{id}\times g_1, C'_1\times G}, \pi_{\mathrm{id}\times g_2, C'_2\times G}].$$

Case 5. If $C = (C'_1 \cup C'_2) \times G$ with $C'_1 \cap C'_2 = \emptyset$, then

$$\pi_{\mathrm{id}\times g, (C'_1\cup C'_2)\times G} = \pi_{\mathrm{id}\times g, C'_1\times G} \circ \pi_{\mathrm{id}\times g, C'_2\times G}.$$

Case 6. If $C = (C'_1 \cup C'_2) \times G$, then

$$\pi_{\mathrm{id}\times g, (C'_1\cup C'_2)\times G} = \pi_{\mathrm{id}\times g, (C'^c_1\cap C'^c_2)^c\times G}.$$

We conclude that $\pi_{\mathrm{id}\times g, C} \in \mathcal{G}_\ell$, and that the provided upper-bounds are correct. □

Note that for any $g \in \mathrm{Sym}(H')$, $g \times \mathrm{id}_G \in \mathrm{Sym}(H)$ is an even permutation since $|G|$ is even. Combining this with the previous lemma, we obtain the following result which allows the conditioning of gates depending on conditions of the form $C' \times G$, and controls their word norm in $\mathcal{G}_\ell$.

LEMMA 4.7. *Let $T : \mathcal{P}(H \times \Gamma^{\ell-1}) \to \mathbb{N}$ be given by the previous lemma. Assume that $g \in \mathrm{Sym}(H')$ is a permutation and $C = C' \times G$ is a $\pi_{g\times\mathrm{id}}$-invariant condition, for some $C' \subseteq H' \times \Gamma^{\ell-1}$. Then the permutation $\pi_{g\times\mathrm{id}, C}$ belongs to $\mathcal{G}_\ell$ with word norm $O(T(C))$.*

We divide the proof of this lemma in three parts.

CLAIM 4.8. *For $g \in \mathrm{Alt}(H' \times G)$ a 3-cycle and $C = C' \times G$ a π_g-invariant condition, the permutation $\pi_{g,C}$ belongs to $\mathcal{G}_\ell$ with word norm $O(T(C))$.*

Proof. Let $g = ((h_1, x_1), (h_2, x_2), (h_3, x_3))$ be a 3-cycle of $\mathrm{Sym}(H' \times G)$ and $C = C' \times G$ be a π_g-invariant condition for some $C' \subseteq H' \times \Gamma^{\ell-1}$.

Consider the condition $B = B' \times G$ where $B' = C' \cap [h_1]_0$, and for $y_1, y_2, y_3 \in G$ three distinct elements, let us define the following permutations:

$$g'_G = (y_1, y_2, y_3) \in \mathrm{Alt}(G),$$
$$g' = ((h_1, y_1), (h_1, y_2), (h_1, y_3)) \in \mathrm{Alt}(H' \times G).$$

By Lemma 4.7, $\pi_{\mathrm{id}\times g'_G, B}$ belongs to $\mathcal{G}_\ell$ with word norm $O(T(B)) = O(T(C))$. However, $\pi_{\mathrm{id}\times g'_G, B} = \pi_{g', C}$, since $B' = C' \cap [h_1]_0$. This proves that $\pi_{g,C}$ belongs to $\mathcal{G}_\ell$ with word norm $O(T(C))$, since g and g' (hence $\pi_{g,C}$ and $\pi_{g',C}$) are conjugated by the involutions $(h_i, x_i) \leftrightarrow (h_1, y_i)$ of $\mathrm{Sym}(H)$. □

CLAIM 4.9. *For $g \in \mathrm{Sym}(H')$ a cycle of support S and $C = C' \times G$ a $\pi_{g\times\mathrm{id}}$-invariant condition, the condition C is $\pi_{g'}$-invariant for every $g' \in \mathrm{Sym}(S \times G)$.*

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Proof. Let $g' \in \mathrm{Sym}(S \times G)$, $(h, x) \in H' \times G$, and $\gamma \in \Gamma^{\ell-1}$. Let us denote $g'(h, x) = (h', x')$. As g is a cycle, there exists k such that $g^k(h) = h'$. Then $\pi_{g'}((h, x) \cdot \gamma) = g'(h, x) \cdot \gamma = (g^k(h), x') \cdot \gamma$, so that

$$\begin{aligned}
& \pi_{g'}((h, x) \cdot \gamma) \in C' \times G \\
\iff\ & (g^k(h), x') \cdot \gamma \in C' \times G \\
\iff\ & \pi_g^k((h, x') \cdot \gamma) \in C' \times G \\
\iff\ & (h, x') \cdot \gamma \in C' \times G \quad \text{as } C' \times G \text{ is } \pi_g\text{-invariant} \\
\iff\ & (h, x) \cdot \gamma \in C' \times G.
\end{aligned}$$

□

We can now conclude the proof.

Proof of Lemma 4.7. Let $g \in \mathrm{Sym}(H')$ and $C = C' \times G$ be a $\pi_{g \times \mathrm{id}}$-invariant condition for some $C' \subseteq H' \times \Gamma^{\ell-1}$. Without loss of generality, we can assume that g is a cycle, whose support we denote as S.

Then, $g \times \mathrm{id}$ belongs to $\mathrm{Alt}(S \times G)$, since G has even cardinality. Additionally, $\mathrm{Alt}(S \times G)$ is generated by its 3-cycles since $|S \times G| \geq 3$. Let us write $g \times \mathrm{id} = c_1 \circ \cdots \circ c_k$ for $c_1, \ldots, c_k$ 3-cycles of $\mathrm{Alt}(S \times G)$. Then C is π_{c_i}-invariant for every c_i by the second claim, so by the first claim, each $\pi_{c_i, C}$ belongs to $\mathcal{G}_\ell$ with word norm $O(T(C))$.

As $\mathrm{Alt}(S \times G) \leq \mathrm{Alt}(H)$ is finite, k is bounded (with bound independent from ℓ), and $\pi_{g \times \mathrm{id}, C}$ is the composition of this bounded number of $\pi_{c_i, C}$. This concludes the proof. □

We finally state an (optional) rephrasing of Lemma 4.7. Readers with a background in complexity theory will find the following version of the statement useful; it is immediate from the definition of the complexity class NC^1.

LEMMA 4.10. *Let $L \subseteq H' \times \Gamma^*$ be a language in NC^1, and $L_n \subseteq H' \times \Gamma^{n-1}$ its words of size n. For any $g \in \mathrm{Alt}(H')$, the conditioned gates $f_{g \times \mathrm{id}, L_n \times G}$ belong to $\mathcal{G}_\ell$ with polynomial word norm in n.*

4.2.2. *Examples.* To clarify Lemma 4.7, we consider several examples: the permutation of adjacent letters, lexicographic comparisons, and (cyclic) ternary additions.

Permuting two adjacent tape-letters

LEMMA 4.11. *Consider the permutation $p \in \mathrm{Sym}(C_\ell)$ of two adjacent tape-letters around the head, that is, the shift-equivariant action on patterns of size 2:*

$$p : \begin{pmatrix} w_0 & w_1 \\ q & \end{pmatrix} \to \begin{pmatrix} w_1 & w_0 \\ q & \end{pmatrix}$$

for any $w = w_0 w_1 \in \Gamma^2$ and $q \in Q$.

Then p belongs to $\mathcal{G}_\ell$ with constant word norm.

Proof. The permutation p is the composition of finitely many commuting $p_{(a,b)}$ that only permutes adjacent tape-letters when they differ and belong to the set $\{a, b\} \subseteq \Gamma$, that is, for $a \neq b \in \Gamma$, the involution $p_{(a,b)} \in \mathrm{Sym}(C_\ell)$ is defined as

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

$$\begin{pmatrix} a & b \\ q & \end{pmatrix} \leftrightarrow \begin{pmatrix} b & a \\ q & \end{pmatrix}$$

for any $q \in Q$. So we only need to prove that every $p_{(a,b)}$ belongs to $\mathcal{G}_\ell$.

Let $a \neq b \in \Gamma$. Define $g_\Gamma \in \mathrm{Sym}(\Gamma)$ as the involution $g_\Gamma = a \leftrightarrow b$. Then $g' = \mathrm{id}_{Q'} \times g_\Gamma$ is a permutation of $\mathrm{Sym}(H')$, and $g = g' \times \mathrm{id}_G$ is a permutation of $\mathrm{Sym}(H)$. By Lemma 4.7, both $\pi_{g,[a]_1 \times G}$ and $\pi_{g,[a]_{-1} \times G}$ belong to $\mathcal{G}_\ell$ with word norm independent of ℓ, and we have $p_{(a,b)} = \Pi$, where

$$\Pi = (\rho^{-1} \circ \pi_{g,[a]_{-1} \times G} \circ \rho) \circ (\pi_{g,[a]_1 \times G}) \circ (\rho^{-1} \circ \pi_{g,[a]_{-1} \times G} \circ \rho)$$

and ρ denotes the right rotation of the head. Indeed, for any $q \in Q$, we calculate

$$\begin{aligned} \Pi\left(\begin{pmatrix} a & b \\ q & \end{pmatrix}\right) &= \begin{pmatrix} b & a \\ q & \end{pmatrix}, \\ \Pi\left(\begin{pmatrix} b & a \\ q & \end{pmatrix}\right) &= \begin{pmatrix} a & b \\ q & \end{pmatrix}, \\ \Pi\left(\begin{pmatrix} a & a \\ q & \end{pmatrix}\right) &= \begin{pmatrix} a & a \\ q & \end{pmatrix}, \end{aligned}$$

and nothing of interest (other than the head moving back and forth) happens on other patterns. □

Remark 4.12. The previous permutation p can be conditioned on any value of the state $q' \in Q'$ by simply replacing the conditions $[a]_1 \times G$ (respectively $[a]_{-1} \times G$) with $([a]_1 \cap [\{q'\} \times \Gamma]_0) \times G$ (respectively $([a]_{-1} \cap [\{q'\} \times \Gamma]_0) \times G$).

Lexicographic comparisons. Let us consider the case of conditions that lexicographically compare the content of the tape with arbitrary fixed words (word equalities and comparisons). By reducing the movement of the head, we obtain better upper bounds than what a strict reading of Lemma 4.6 would suggest.

LEMMA 4.13. *Given a lexicographic equality/comparison* $\sim \in \{=, <, \leq, >, \geq\}$ *and a word* $c \in \{0, 1, 2\}^l$ *for some* $l \leq \ell$*, consider a condition* $C \subseteq H \times \Gamma^{\ell-1}$ *of the form* $\sim c$*, that is,*

$$\begin{pmatrix} a_0 & \dots & a_{l-1} & a_l & \dots & a_{\ell-1} \\ q & & & & & \end{pmatrix} \in C \iff a_0 \cdots a_{l-1} \sim c_0 \cdots c_{l-1}.$$

Then $T(C) = O(|c|^2)$.

Proof. To prove this result, we use a divide and conquer approach: given $l \leq \ell$, any integer $l' \leq l$, and two words $w \in \{0, 1, 2\}^l$ and $c \in \{0, 1, 2\}^l$,

$$\begin{aligned} w = c &\iff (w_{[\![0,l'-1]\!]} = c_{[\![0,l'-1]\!]}) \wedge (w_{[\![l',l-1]\!]} = c_{[\![l',l-1]\!]}), \\ w < c &\iff (w_{[\![0,l'-1]\!]} < c_{[\![0,l'-1]\!]}) \\ &\qquad \vee ((w_{[\![0,l'-1]\!]} = c_{[\![0,l'-1]\!]}) \wedge (w_{[\![l',l-1]\!]} < c_{[\![l',l-1]\!]})). \end{aligned}$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

So if C is the condition $= c$ and $g_1, g_2 \in \mathrm{Alt}(G)$, then

$$\pi_{\mathrm{id}\times[g_1,g_2],=c} = [\pi_{\mathrm{id}\times g_1,=c_{[\![0,l'-1]\!]}},\ \rho^{-l'} \circ \pi_{\mathrm{id}\times g_2,=c_{[\![l',l-1]\!]}} \circ \rho^{l'}].$$

Taking $l' = \lfloor l/2 \rfloor$ and iterating, we obtain $T(C) = O(l^2) = O(|c|^2)$. Similarly, if C is the condition $< c$, we obtain $T(C) = O(|c|^2)$. The other cases follow using elementary Boolean algebra on the operators of $\{=, <, \leq, >, \geq\}$. □

Remark 4.14. The previous lemma can still be used to compare words w and c at non-contiguous positions. Denote by $\bullet$ a wildcard symbol that represents positions which will be ignored when performing the comparison $w \sim c$, that is, for any two words $w_0 \cdots w_{l-1} \in \{0, 1, 2\}^l$ and $c_0 \cdots c_{l-1} \in (\{0, 1, 2\} \cup \{\bullet\})^l$, if $i_0 < \cdots < i_n$ are the non-wildcard positions in c (that is, $c_i \neq \bullet$ if and only if $i \in \{i_0, \ldots, i_n\}$), we say that $w \sim c$ if the usual lexicographic comparison $w_{i_0} \cdots w_{i_n} \sim c_{i_0} \cdots c_{i_n}$ is true.

Then the previous lemma still holds for comparisons $w \sim c$ for $w \in \{0, 1, 2\}^l$ and $c \in (\{0, 1, 2\} \cup \{\bullet\})^l$.

(Cyclic) ternary addition. Finally, let us consider ternary additions, that is, adding the number $v_3(k)$ for $k \in \{0, 1, 2\}^*$ to the $|k|$ letters on the right side of the head by considering them as a counter in base 3.

LEMMA 4.15. *For $k \in \{0, 1, 2\}^*$, $|k| \leq \ell$, let $\mathrm{add}_k \in \mathrm{Sym}(C_\ell)$ be defined as the following shift-equivariant action:*

$$\mathrm{add}_k : \begin{pmatrix} a_0 & \ldots a_{|k|-1} & a_{|k|} & \ldots a_{\ell-1} \\ q & & & \end{pmatrix} \to \begin{pmatrix} a'_0 & \ldots a'_{|k|-1} & a_{|k|} & \ldots a_{\ell-1} \\ q & & & \end{pmatrix},$$

where $v_3(a'_0 \cdots a'_{|k|-1}) = v_3(a_0 \cdots a_{|k|-1}) + v_3(k) \bmod 3^{|k|}$.

Then add_k belongs to $\mathcal{G}_\ell$ with word norm $O(|k|^3)$.

Proof. Let $r = \mathrm{id}_Q \times (0, 1, 2) \in \mathrm{Sym}(Q \times \Gamma)$ denote the rotation of the tape-letter by the 3-cycle $(0, 1, 2) \in \mathrm{Alt}(\Gamma)$. To perform additions modulo $3^{|k|}$, we apply the standard 'school algorithm', whose main difficulty consists in performing carries.

The application of a carry only needs to be performed when the addition of the previously added digits has overflowed, that is, when the digits on which the addition has already been performed (from right to left) have a smaller value than the rightmost digits of k. This is exactly what we do.

First, move the head to the right of the counter by applying $\rho^{|k|-1}$. Then, apply either π_r if the last digit of k is 1 (that is, $k_{|k|-1} = 1$), or π_{r^2} if $k_{|k|-1} = 2$, or nothing if $k_{|k|-1} = 0$.

Then, for $j \in [\![1, |k| - 2]\!]$, do the following.

(1) Move the head to the left with ρ^{-1}.
(2) Apply either π_r if $k_{|k|-j-1} = 1$, or π_{r^2} if $k_{|k|-j-1} = 2$, or nothing if $k_{|k|-j-1} = 0$.
(3) Perform the carry: denoting $k' = k_{[\![|k|-j,|k|-1]\!]}$, apply $\pi_{r,<\bullet k'}$ where $\bullet$ is the wildcard symbol, using the notation of Lemma 4.13 and Remark 4.14.

Then add_k is the composition of $O(|k|)$ permutations of $\mathcal{G}_\ell$, each having word norm $O(|k|^2)$ (according to Lemma 4.13). This concludes the proof. □

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Remark 4.16. The addition add_k can be conditioned on any value of the state $q' \in Q'$ by simply intersecting all the conditions for applying π_r (or π_{r^2}) with $[\{q'\} \times \Gamma]_0 \times G$.

4.2.3. *The ducking trick.* Recall that the set Q splits into $Q = Q' \times G$, and that Q' itself splits into $Q' = Q^{(\mathrm{o})} \times D$: the states $Q^{(\mathrm{o})} = \{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\} \times \{1, 2\}$ of the original SMART machine, and the duck $D = \{d_1, d_2\}$. Recall that for $S \subseteq C_\ell$, we denote by $S[d_1]$ (respectively $S[d_2]$) the subset of tapes in S containing a head whose duck is $d = d_1$ (respectively $d = d_2$).

In this section, we introduce the *ducking trick*: this method uses this ducking component D to realize piecewise-defined bijections, in particular in the proof of Lemma 4.3 that follows.

Let $f = f^{(\mathrm{o})} \times \mathrm{id}_{D \times G} : Q \times \Gamma^\ell \to Q \times \Gamma^\ell$ be a map defined on patterns of length ℓ by some $f^{(\mathrm{o})} : Q^{(\mathrm{o})} \times \Gamma^\ell \to Q^{(\mathrm{o})} \times \Gamma^\ell$. Assume that f is a piecewise-defined bijection, that is, that there exist partitions $\sqcup_{i=1}^p U_i^{(\mathrm{o})}$ and $\sqcup_{i=1}^p V_i^{(\mathrm{o})}$ of $Q^{(\mathrm{o})} \times \Gamma^\ell$ and maps $f_i^{(\mathrm{o})} : U_i^{(\mathrm{o})} \to V_i^{(\mathrm{o})}$ such that $f^{(\mathrm{o})} = \bigsqcup f_i^{(\mathrm{o})}$ (as the union of graphs of functions).

In what follows, we denote $U_i = U_i^{(\mathrm{o})} \times D \times G \subseteq Q \times \Gamma^\ell$ and $V_i = V_i^{(\mathrm{o})} \times D \times G \subseteq Q \times \Gamma^\ell$, and $f_i = f_i^{(\mathrm{o})} \times \mathrm{id} : U_i \to V_i$, so that $f = \bigsqcup_i f_i$.

Assume the following.

(1) There exist maps $g_i \in \mathcal{G}_\ell$ with polynomial word norm in ℓ such that g_i agrees with f_i on $U_i[d_2]$ (that is, $g_i|_{U_i[d_2]} = f_i$), and is the identity on $U_i[d_1]$ (that is, $g_i|_{U_i[d_1]} = \mathrm{id}$).

(2) Each U_i has a simple description. Specifically, we should have a Boolean circuit of type NC^1 for it.

In our application, the descriptions of the U_i involve only numerical comparisons and direct letter comparisons, and the simplicity is encapsulated in the formulas in §4.2.2. The maps g_i differ from f_i in the following way: instead of having domain U_i, they act on the whole set of patterns of length ℓ, and are only required to act on $U_i[d_2]$ as f_i does (and be the identity on $U_i[d_1]$).

CLAIM 4.17. (Ducking trick) *Consider* $F : Q \times \Gamma^\ell \to Q \times \Gamma^\ell$ *defined as*

$$F(w) = \begin{cases} f(w) & \text{if } w \in Q \times \Gamma^\ell[d_1], \\ f^{-1}(w) & \text{if } w \in Q \times \Gamma^\ell[d_2], \\ w & \text{otherwise.} \end{cases}$$

Under the assumptions above, F *belongs to* $\mathcal{G}_\ell$ *with polynomial word norm in* ℓ.

Proof. Denote by $d' \in \mathrm{Sym}(D)$ the involution $d' = d_1 \leftrightarrow d_2$. Then $d = \mathrm{id}_{Q^{(\mathrm{o})}} \times d' \times \mathrm{id}_G \times \mathrm{id}_\Gamma \in \mathrm{Sym}(H)$ is an involution. As the sets U_i are π_d-invariant, by Lemma 4.7, each π_{d,U_i} belongs to $\mathcal{G}_\ell$ with word norm $O(T(U_i))$ (which is polynomial in ℓ, because of the NC^1-description).

Now conjugate the permutations π_{d,U_i} by their respective g_i^{-1}, to get

$$F_i = (\pi_{d,U_i})^{(g_i^{-1})} = g_i \circ \pi_{d,U_i} \circ g_i^{-1}.$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

This is an involution that acts like f_i on $U_i[d_1]$ (and flips the duck), like f_i^{-1} on $V_i[d_2]$ (and flips the duck), and is the identity on the rest of $Q \times \Gamma^\ell$. Indeed, π_{d,U_i} exchanges words $w \in U_i[d_1]$ with $\pi_d(w) \in U_i[d_2]$. Conjugating it with g_i^{-1} effectively means that we translate the domain of this permutation by g_i. More precisely, this is the involution

$$\begin{aligned} F_i : g_i(U_i[d_1]) &\leftrightarrow g_i(U_i[d_2]), \\ g_i(w) &\leftrightarrow g_i(\pi_d(w)). \end{aligned}$$

As g_i acts like f_i on $U_i[d_2]$ and is the identity on $U_i[d_1]$, we have in fact:

$$\begin{aligned} F_i : U[d_1] &\leftrightarrow V_i[d_2], \\ w &\leftrightarrow f_i(\pi_d(w)). \end{aligned}$$

Additionally, F_i has polynomial word norm because g_i and π_{d,U_i} have polynomial word norm.

Finally, each time F_i acts non-trivially, it also flips the duck, so all the F_i have disjoint support. Composing them in any arbitrary order, we obtain that $F = \pi_d \circ \prod_i F_i$ belongs to $\mathcal{G}_\ell$ with polynomial word norm in ℓ. □

4.3. *Implementing encodings and additions*

4.3.1. *Proof of Lemma 4.3.* In this section, we use the previous lemmas to implement the inductive encoding of §3.4 in $\mathcal{G}_\ell$. More precisely, recall the statement of Lemma 4.3.

LEMMA 4.3. *Let $E_\ell : C_\ell \to C_\ell$ be the encoding map defined in §3.2. There exists some $\tilde{E}_\ell : C_\ell \to C_\ell$ in $\mathcal{G}_\ell$ with word norm $O(\ell^4)$ such that*

$$w \in C_\ell[d_1] \implies \tilde{E}_\ell(w) = E_\ell(w),$$

where $C_\ell[d_1]$ is the set of cyclic tapes $w \in C_\ell$ having a head with duck $d = d_1$.

This section is dedicated to the proof of this lemma. Informally, we use the inductive decomposition of E_ℓ into

$$E_\ell = F_{\ell,\text{final}} \circ \left(\prod_{k=0}^{\ell-2} F_{k\to k+1} \right) \circ F_{\text{init}}$$

defined in §3.4, and build every F_{init}, $F_{k\to k+1}$, and $F_{\ell,\text{final}}$ in the group $\mathcal{G}_\ell$. Each of these steps being defined as piecewise-defined bijections, we use the *ducking trick* mentioned in §4.2.3: with the duck $D = \{d_1, d_2\}$, we write each bijection as a product of involutions that swap ducks d_1 and d_2.

Let F be one of F_{init}, $F_{k\to k+1}$, or $F_{\ell,\text{final}}$. Here, F rewrites patterns as explained in §3.4, and copies all other symbols unchanged. Let $U \overset{F}{\mapsto} V$ be one of the cases of F, that is, one of the rules of pattern rewriting that defines F. (Apart from the proof of Lemma 4.19, there is no need to actually know what U, V, and F precisely are to understand the following statements.)

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Define

$$\begin{aligned}\tilde{F}_{U,V} : U[d_1] &\leftrightarrow V[d_2]\\ w &\leftrightarrow \pi_d(F(w))\end{aligned}$$

as the involution of $\mathrm{Sym}(C_\ell)$ that swaps words $w \in U[d_1]$ with words $\pi_d(F(w)) \in V[d_2]$, where π_d is the duck-flipping involution. The involution $\tilde{F}_{U,V}$ applies F forward on tapes with duck d_1 and swaps the duck, and applies F backwards on tapes with duck d_2 and swaps the duck.

LEMMA 4.18. *For any such case $U \overset{F}{\mapsto} V$, the permutation $\tilde{F}_{U,V}$ belongs to $\mathcal{G}_\ell$ with norm $O(\ell^3)$.*

The proof relies on the structure of the different cases $U \overset{F}{\mapsto} V$ defined in §3.4.

Sketch of a proof. Let $U \overset{F}{\mapsto} V$ be such a piece.

The condition U checks the state of the head, the value of a few distinct tape-letters, and a counter being non-zero (in the case of Figure 4) or full-zero (in the case of Figure 5). Denoting by $d = \mathrm{id}_{Q^{(\circ)}} \times d' \times \mathrm{id}_G \times \mathrm{id}_\Gamma \in \mathrm{Sym}(H)$ for $d' : d_1 \leftrightarrow d_2 \in \mathrm{Sym}(D)$ the involution that swaps the duck, the gate $\pi_{d,U}$ conditioned on U belongs to $\mathcal{G}_\ell$ with norm $O(\ell^2)$.

By conjugating $\pi_{d,U}$ by a sequence of permutations conditioned on the duck being d_2, we can then build $\tilde{E}_{U,V}$ with norm $O(\ell^3)$ in $\mathcal{G}_\ell$. □

Example 4.19. On an example, consider $U \overset{F}{\mapsto} V$ to be the following transformation between levels k and $k+1$ (this is the third rewrite in Figure 4):

$$\begin{pmatrix} * & c & \overset{1}{\underset{\blacktriangleright_1}{}} & * \end{pmatrix} \rightarrow \begin{pmatrix} * & \overset{[\mathrm{v}_3(c)+f(n)+1]_{(3)}}{\underset{\blacktriangleleft_1}{}} & * \end{pmatrix}$$

Then the condition U is the conjunction of the head being in state $\blacktriangleright_1$, on top of the tape-letter 1, and the $k+1$ tape-letters on its left being non-zero. By Lemma 4.13, $\pi_{d,U}$ belongs to $\mathcal{G}_\ell$ with norm $O(k^2)$.

To obtain $\tilde{E}_{U,V}$, we then conjugate $\pi_{d,U}$ with the sequence of *inverses* of the following permutations, all of which are *conditioned on the head having duck d_2*.

(1) At first, when $\pi_{d,U}$ is conjugated by nothing (that is, the identity), we have

$$\begin{pmatrix} * & c & \overset{1}{\underset{\blacktriangleright_1^{d_1}}{}} & * \end{pmatrix} \leftrightarrow \begin{pmatrix} * & c & \overset{1}{\underset{\blacktriangleright_1^{d_2}}{}} & * \end{pmatrix}$$

and, again, this permutation is only applied if c is a non-zero counter. (Note that we see here the full support of the permutation.)

(2) Let $g' = (1,0) \in \mathrm{Sym}(\Gamma)$, and $g = \mathrm{id}_{Q^{(\circ)}} \times \{d_2\} \times \mathrm{id}_G \times g' \in \mathrm{Sym}(H)$. Apply the gate π_g, which has word norm $O(1)$ in $\mathcal{G}_\ell$. At the moment, we have the permutation:

$$\begin{pmatrix} * & c & \overset{1}{\underset{\blacktriangleright_1^{d_1}}{}} & * \end{pmatrix} \leftrightarrow \begin{pmatrix} * & c & \overset{0}{\underset{\blacktriangleright_1^{d_2}}{}} & * \end{pmatrix}$$

(3) Using the permutation p from Lemma 4.11 (that exchanges two adjacent letters) conditioned on the duck being $d = d_2$ (see Remark 4.12), and moving the head using

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

the generator ρ of $\mathcal{G}_\ell$, we move the counter c one step to its right and the tape-letter 0 under the head $k+1$ steps to its left. At the moment, we have the permutation:

$$\left(* \quad c \quad \begin{matrix} 1 \\ \blacktriangleright_1^{d_1} \end{matrix} \quad * \right) \leftrightarrow \left(* \quad \begin{matrix} 0 \\ \blacktriangleright_1^{d_2} \end{matrix} \quad c \quad * \right)$$

(4) For $w \in \{0,1,2\}^{k+2}$ such that $v_3(w) = f(k)+1$, we apply add_w from Lemma 4.15 (namely, the permutation that adds $f(k)+1$ to the ternary number of length $k+2$ to the right of the head) conditioned on the duck being $d = d_2$ (see Remark 4.16). It has word norm $O(k^3)$. Then, apply ρ^{-1} on ducks $d = d_2$. At the moment, we have the permutation:

$$\left(* \quad c \quad \begin{matrix} 1 \\ \blacktriangleright_1^{d_1} \end{matrix} \quad * \right) \leftrightarrow \left(\begin{matrix} * \\ \blacktriangleright_1^{d_2} \end{matrix} \quad [\mathrm{v}_3(c) + f(k) + 1]_{(3)} \quad * \right)$$

(5) Let $g' = \blacktriangleright_1^{d_2} \leftrightarrow \blacktriangleleft_1^{d_2} \in \mathrm{Sym}(Q)$ and $g = g' \times \mathrm{id}_\Gamma \in \mathrm{Sym}(H)$. We apply π_g and finally obtain $\tilde{F}_{U,V}$:

$$\left(* \quad c \quad \begin{matrix} 1 \\ \blacktriangleright_1^{d_1} \end{matrix} \quad * \right) \leftrightarrow \left(\begin{matrix} * \\ \blacktriangleleft_1^{d_2} \end{matrix} \quad [\mathrm{v}_3(c) + f(k) + 1]_{(3)} \quad * \right)$$

Having built each case of the piecewise-defined function F, we complete the 'ducking' process and obtain an immediate corollary.

LEMMA 4.20. *For F being either F_{init}, $F_{k\to k+1}$, or $F_{\ell,\mathrm{final}}$, define*

$$\tilde{F}(w) = \begin{cases} F(w) & \text{if } w \in C_\ell[d_1], \\ F^{-1}(w) & \text{if } w \in C_\ell[d_2], \\ w & \text{otherwise.} \end{cases}$$

Then $\tilde{F}$ belong to $\mathcal{G}_\ell$ with word norm $O(\ell^3)$.

Proof. Each of these transformations F is composed of finitely many cases $U \overset{F}{\mapsto} V$, and by the previous lemma, each $\tilde{F}_{U,V}$ belong to $\mathcal{G}_\ell$ with word norm $O(\ell^3)$. Denote by $d \in \mathrm{Sym}(H)$ the involution that swaps ducks $d = d_1$ and $d = d_2$. As the $\tilde{F}_{U,V}$ commute (since they have disjoint support), the involution $\tilde{F}$ can be written as the composition of finitely many $\tilde{F}_{U,V}$ and π_d. □

We then conclude and obtain Lemma 4.3.

Proof of Lemma 4.3. Recall that the encoding map E_ℓ from §3.2 is defined as $E_\ell = F_{\ell,\mathrm{final}} \circ \prod_{k=0}^{\ell-2} F_{k\to k+1} \circ F_{\mathrm{init}}$.

By the previous lemma, there exists $\tilde{F}_{\mathrm{init}}$, $\tilde{F}_{k\to k+1}$, and $\tilde{F}_{\ell,\mathrm{final}}$ in $\mathcal{G}_\ell$ with word norm $O(\ell^3)$ that agree on $C_\ell[d_1]$ with their respective $F_{\ell,\mathrm{final}}$, $F_{k\to k+1}$, and F_{init}. As the set $C_\ell[d_1]$ is stable by these permutations, we define

$$\tilde{E}_\ell = \tilde{F}_{\ell,\mathrm{final}} \circ \prod_{k=0}^{\ell-2} \tilde{F}_{k\to k+1} \circ \tilde{F}_{\mathrm{init}}.$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Then the map $\tilde{E}_\ell$ satisfies:

$$w \in C_\ell[d_1] \implies \tilde{E}_\ell(w) = E_\ell(w) \in C_\ell[d_1]. \qquad \square$$

One should note that, on configurations $w \in C_\ell[d_2]$, $\tilde{E}_\ell$ 'produces garbage': we claim no meaningful interpretation for the image of such w. In particular, we note that while for $w \in C_\ell[d_2]$, we have $\tilde{F}_{\text{init}}(w) = F^{-1}_{\text{init}}(w)$, the composition $F^{-1}_{\ell,\text{final}} \circ \prod_{k=0}^{\ell-2} F^{-1}_{k\to k+1} \circ F^{-1}_{\text{init}}$ does *not* equal the inverse of E.

4.3.2. *Proof of Lemma 4.4.* Let us recall the statement of Lemma 4.4.

LEMMA 4.4. *Let $(+n)_\ell$ be the bijection of C_ℓ that performs the addition of $n \in \mathbb{N}$ in base $2\ell \cdot 3^\ell$ on the orbits positions encoded by E_ℓ. Recall that $(+n/_{d_1})_\ell : C_\ell \to C_\ell$ is defined as*

$$(+n/_{d_1})_\ell(w) = \begin{cases} (+n)_\ell(w) & \text{if } w \in C_\ell[d_1], \\ w & \text{otherwise.} \end{cases}$$

Then $(+n/_{d_1})_\ell$ belongs to $\mathcal{G}_\ell$ with word norm $O(\ell^3)$.

More precisely, denoting by q_b any state in $[q_b] = \{q_b\} \times D \times G$, the map $(+n/_{d_1})_\ell$ bijectively acts on the following patterns of length ℓ as

$$+n/_{d_1}\begin{pmatrix} c_0 & \cdots & c_{\ell-1} \\ q_b & & \end{pmatrix} \to \sigma^{-\varepsilon a}\left(\begin{pmatrix} c'_0 & \cdots & c'_{\ell-1} \\ q_{b'} & & \end{pmatrix}\right)$$

if $q_b \in \{q_b\} \times \{d_1\} \times G$, where $b' \cdot c' \in \{1,2\} \cdot \{0,1,2\}^\ell$ encodes $((b-1)\cdot 3^\ell + v_3(c)) + n \bmod 2 \cdot 3^\ell$, a is the quotient of $((b-1)\cdot 3^\ell + v_3(c)) + n$ by $2 \cdot 3^\ell$, and $\varepsilon = +1$ if $q \in \{\blacktriangleright, \triangleright\}$, $\varepsilon = -1$ if $q \in \{\blacktriangleleft, \triangleleft\}$, and is the identity otherwise.

We now prove Lemma 4.4.

Proof. Using the permutation p of constant word norm from Lemma 4.11, conditioned on the duck being $d = d_1$ and the state being either in $\{\blacktriangleright, \triangleright\}$ or in $\{\blacktriangleleft, \triangleleft\}$ (see Remark 4.12), we can rotate the tape either right (if $\blacktriangleright$ or $\triangleright$) or left (if $\blacktriangleleft$, $\triangleleft$) at most ℓ times with word norm $O(\ell^2)$. We can now assume that $0 \le n < 2 \cdot 3^\ell$.

Let $b \cdot k \in \{1,2\} \cdot \{0,1,2\}^\ell$ encode n, that is, $v_3(k) = n \bmod 3^\ell$, and $b-1$ be the quotient of n by 3^ℓ. We first add $b \cdot k$ without rotating the tape.

To do so, apply the permutation add_k of Lemma 4.15 to add $n \bmod 3^\ell$ to the cyclic tape conditioned on $d = d_1$ (see Remark 4.16). Then, very similarly to the proof of the same lemma, we perform a carry in the component $\{1,2\}$ of the state (denoting $s = s' \times \{d_1\} \times \text{id}_G \times \text{id}_\Gamma \in \text{Sym}(H)$, where $s' = (q,1) \leftrightarrow (q,2) \in \text{Sym}(Q^{(\text{o})})$, we perform the conditioned gate $\pi_{s,<k}$ defined in Lemma 4.13), and depending on the bit b, we cyclically rotate the bit carried by state (if $b = 2$, we apply π_s, and if $b = 1$, we apply the identity).

We are now left with performing a cyclic rotation of the whole tape if the addition of n in the previous paragraph overflowed. Let $s_1 \in \text{Sym}(H)$ and $s_2 \in \text{Sym}(H)$ respectively flip filled arrows $\blacktriangleright \leftrightarrow \blacktriangleleft$ and $\triangleright \leftrightarrow \triangleleft$, that is, $s_1 = s'_1 \times \{d_1\} \times \text{id} \times \text{id} \in \text{Sym}(H)$ (respectively $s_2 = s'_2 \times \{d_1\} \times \text{id} \times \text{id} \in \text{Sym}(H)$), for $s'_1 = (\blacktriangleright \leftrightarrow \blacktriangleleft) \times \text{id} \in \text{Sym}(Q^{(\text{o})})$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

(respectively $s_2' = (\triangleright \leftrightarrow \triangleleft) \times \mathrm{id} \in \mathrm{Sym}(Q^{(\mathrm{o})})$). By Lemma 4.13, both $\pi_{s_1,C}$ and $\pi_{s_2,C}$ belong to $\mathcal{G}_\ell$ with word norm $O(\ell^2)$, for C the overflowing condition $C = <(b \cdot k)$. (We slightly abuse notation here when writing $<(b \cdot k)$: these conditions were defined for comparisons of length $\leq \ell$ of tape-letters, here the comparison is of length $\ell + 1$. Instead of just comparing on the tape, the bit $\{1, 2\}$ carried in the state is also part of the comparison.)

Let $r_\blacktriangleright \in \mathrm{Sym}(C_\ell)$ (respectively $r_\triangleright \in \mathrm{Sym}(C_\ell)$) be the right cyclic shift of whole tapes having states in $\{\blacktriangleright\} \times \{1, 2\} \times \{d_1\} \times G$ (respectively $\{\triangleright\} \times \{1, 2\} \times \{d_1\} \times G$). These permutations belong to $\mathcal{G}_\ell$ with word norm $O(\ell)$ by Lemma 4.11.

Then the commutator $[\pi_{s_1,C}, r_\blacktriangleright]$ (resp. $[\pi_{s_2,C}, r_\triangleright]$) shifts the whole tape if and only if an overflow happened, and it either shifts to the right if the state is in $\{\blacktriangleright\} \times \{1, 2\} \times \{d_1\} \times G$ (respectively $\{\triangleright\} \times \{1, 2\} \times \{d_1\} \times G$), or to the left if the state is in $\{\blacktriangleleft\} \times \{1, 2\} \times \{d_1\} \times G$ (respectively $\{\triangleleft\} \times \{1, 2\} \times \{d_1\} \times G$).

We conclude that $+n/_{d_1}$, being the composition of the previous paragraphs, belongs to $\mathcal{G}_\ell$ with word norm $O(\ell^3)$. □

5. *Distortion of automorphisms of the full shift*

This chapter contains a proof of Theorem A.

THEOREM A. *For any non-trivial alphabet A, the group* $\mathrm{Aut}(A^{\mathbb{Z}})$ *has an element g of infinite order such that* $|g^n|_F = O(\log^4 n)$ *for some finite set F.*

By [33], the automorphism groups of full shifts with different (non-trivial) alphabets embed in each other, so we only need to prove that *there exists* Σ *and some* $g \in \mathrm{Aut}(\Sigma^{\mathbb{Z}})$ *of infinite order such that* $|g^n|_F = O(\log^4 n)$ *for some finite set F.*

We first introduce the conveyor belt construction, which allows to embed a given Turing machine into a full shift. We then define a group $\mathcal{G}_*$, which contains every Turing machine on a given set of states and alphabet, along with some instructions to modify the conveyor belt structures in configurations. Then, we state Lemma 5.1: every finitarily distorted Turing machine $\mathcal{M}$ gives rise to a distorted automorphism in $\mathcal{G}_*$. Overall, the proof consists in transporting the already existing (finitary) distortion into a full shift.

We finally add some various optimizations in the case of the SMART machine. All our results are summarized in Lemma 5.14, which precisely defines a full shift $\Sigma^{\mathbb{Z}}$ and a distorted automorphism f_S of infinite order which satisfies the aforementioned polylogarithmic word norm of degree four.

5.1. *Context and statements of results*

5.1.1. *Conveyor belts.* Recall that a Turing machine $\mathcal{M} = (Q, \Gamma, \Delta)$ acts on the related subshift $\mathcal{X}$ (see §2), that is, on the set of bi-infinite tapes containing at most one head (that is, one symbol of $Q \times \Gamma$).

To make a Turing machine $\mathcal{M} = (Q, \Gamma, \Delta)$ act on a full shift instead, we use the conveyor belt trick (see for example [28, Lemma 3]). Let

$$\Sigma = (\Gamma^2 \times \{+1, -1\}) \sqcup ((Q \times \Gamma) \times \Gamma) \sqcup (\Gamma \times (Q \times \Gamma)),$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

where we call *conveyor bits* the bits of $\{+1, -1\}$ in $(\Gamma^2 \times \{+1, -1\})$, and define the action of $\mathcal{M}$ on $\Sigma^{\mathbb{Z}}$ as the following automorphism $f_{\mathcal{M}}$.

Any configuration $x \in \Sigma^{\mathbb{Z}}$ uniquely splits into $x = \cdots w_{-2}w_{-1}w_0w_1w_2 \cdots$ such that for every $i \in \mathbb{Z}$, we have either

$$w_i \in (\Gamma^2 \times \{+1\})^*(((Q \times \Gamma) \times \Gamma) \cup (\Gamma \times (Q \times \Gamma)))(\Gamma^2 \times \{-1\})^*$$
$$\text{or } w_i \in (\Gamma^2 \times \{+1\})^+(\Gamma^2 \times \{-1\})^+$$

with the exception that on some configurations, there might exist a leftmost or rightmost word with an infinite number of $+1$ or -1. (The corresponding decomposition claim in [28] has a misprint, as it uses Kleene stars also in the second form.) We describe the action of $f_{\mathcal{M}}$ on such finite words $w = w_i$ of these two forms: as configurations made of infinitely many finite w are dense and $f_{\mathcal{M}}$ will be uniformly continuous on these, $f_{\mathcal{M}}$ will uniquely extend to an automorphism on the full shift.

- On words $w \in (\Gamma^2 \times \{+1\})^+(\Gamma^2 \times \{-1\})^+$, we do nothing.
- On words $w \in (\Gamma^2 \times \{+1\})^*(((Q \times \Gamma) \times \Gamma) \cup (\Gamma \times (Q \times \Gamma)))(\Gamma^2 \times \{-1\})^*$, let

$$w' \in (\Gamma^2)^*(((Q \times \Gamma) \times \Gamma) \cup (\Gamma \times (Q \times \Gamma)))(\Gamma^2)^*$$

be the word obtained by erasing the conveyor bits $+1$ and -1 from w. We see w' as a conveyor belt of length $2|w|$, that is, the superimposition of a top word u and a bottom word v, glued together at their borders as if the words were laid down on a conveyor belt.

More precisely, for π_1 and π_2 the projections to the first and second component of the alphabet, let $u = \pi_1(w')$ and $v = \overline{\pi_2(w')}$, the reverse of $\pi_2(w')$. One of these words is in Γ^+, and the other is in $\Gamma^*(Q \times \Gamma)\Gamma^*$. We make $\mathcal{M}$ act on $(uv)^{\mathbb{Z}}$ (despite it having infinitely many heads, it should be clear what this means, as all the heads move with the same transition), that is, we define

$$u'v' = T_{\mathcal{M}}((uv)^{\mathbb{Z}})_{[\![0,2|w|-1]\!]}.$$

Note that $u'v'$ also contains exactly one head. We then rewrap $u'v'$ into a conveyor belt of $(\Gamma^2)^*(((Q \times \Gamma) \times \Gamma) \cup (\Gamma \times (Q \times \Gamma)))(\Gamma^2)^*$, and add conveyor bits $+1$ (respectively -1) to the cell symbols in Γ^2 to the left of the head (respectively right).

This defines how $f_{\mathcal{M}}$ acts on such words w. This can be summarized as $f_{\mathcal{M}}$ considers such words as a cyclic tape folded in the shape of a conveyor belt, and acts on the cyclic tape.

Note that x and $f_{\mathcal{M}}(x)$ have the same decomposition into a product of conveyor belts, and that if $\mathcal{M}$ is reversible, then $f_{\mathcal{M}}$ is an automorphism of $\mathrm{Aut}(\Sigma^{\mathbb{Z}})$.

5.1.2. *Group of Turing machines on conveyor belts.* Similar to the group of Turing machines $\mathcal{G}_\ell$ acting on the finite cyclic tapes of $\mathcal{C}_\ell$, let us define another point of view on Turing machines acting on full shifts. For $g \in \mathrm{Sym}(Q \times \Gamma)$, define f_g^{up}, f_g^{down}, $f_g \in \mathrm{Aut}(\Sigma^{\mathbb{Z}})$ as

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

$$f_g^{\mathrm{up}}(x)_i = \begin{cases} x_i & \text{if } x_i \in \Gamma^2 \times \{+1, -1\} \text{ or } x_i \in (\Gamma \times (Q \times \Gamma)), \\ (g(a), b) & \text{if } x_i = (a, b) \in ((Q \times \Gamma) \times \Gamma), \end{cases}$$

$$f_g^{\mathrm{down}}(x)_i = \begin{cases} x_i & \text{if } x_i \in \Gamma^2 \times \{+1, -1\} \text{ or } x_i \in ((Q \times \Gamma) \times \Gamma), \\ (a, g(b)) & \text{if } x_i = (a, b) \in (\Gamma \times (Q \times \Gamma)), \end{cases}$$

$$f_g(x)_i = \begin{cases} x_i & \text{if } x_i \in \Gamma^2 \times \{+1, -1\}, \\ (g(a), b) & \text{if } x_i = (a, b) \in ((Q \times \Gamma) \times \Gamma), \\ (a, g(b)) & \text{if } x_i = (a, b) \in (\Gamma \times (Q \times \Gamma)), \end{cases}$$

for x a configuration of $\Sigma^{\mathbb{Z}}$.

For every $q \in Q$, define ρ_q as the right movement of heads in state q *inside their own conveyor belts*, and $\rho = \prod_{q \in Q} \rho_q$. Then define $\mathcal{G}$ the group they generate:

$$\mathcal{G} = \langle \{f_g^{\mathrm{up}}, f_g^{\mathrm{down}}, f_g \mid g \in \mathrm{Sym}(Q \times \Gamma)\} \cup \{\rho_q \mid q \in Q\} \rangle.$$

The generators of $\mathcal{G}_\ell$ introduced in §4 are in direct correspondence with the generators $\{f_g \mid g \in \mathrm{Sym}(Q \times \Gamma)\}$ and $\{\rho_q \mid q \in Q\}$ of $\mathcal{G}$, and the latter can also be seen as the basic instructions of Turing machines: moving heads based on their states or permuting their values.

In §4, we mentioned that every Turing machine $\mathcal{M} = (Q, \Gamma, \ldots)$ belongs to the groups $\mathcal{G}_\ell$. Similarly, for any such machine, the automorphism $f_{\mathcal{M}}$ belongs to $\mathcal{G}$, since it is the composition $\beta_{-1} \circ \beta_{+1} \circ f_\alpha$:

$$\alpha(q, a) = \begin{cases} (q', b) & \text{if } (q, a, q', b) \in \Delta, \\ (q', a) & \text{if } (q, \pm 1, q') \in \Delta, \end{cases}$$

$$\beta_{+1} = \prod_{q' \mid \text{ there exists } q, (q, +1, q') \in \Delta} \rho_{q'},$$

$$\beta_{-1} = \prod_{q' \mid \text{ there exists } q, (q, -1, q') \in \Delta} {\rho_{q'}}^{-1}.$$

5.1.3. *Decorating Turing machines.* Given any two sets Γ_0 and Q_0, define

$$\Gamma = \Gamma_0,$$
$$Q = Q_0 \times D \times G,$$
$$\Sigma = (\Gamma^2 \times \{+1, -1\}) \sqcup ((Q \times \Gamma) \times \Gamma) \sqcup (\Gamma \times (Q \times \Gamma)),$$

where $D = \{d_\rightarrow, d_\leftarrow\}$ and $G = [\![0, 5]\!]$. As in §4.1.1, D is called the *duck* and G is called the *ghost*.

We say that Γ and Q are the decorated versions of Γ_0 and Q_0.

5.1.4. *Generalized group $\mathcal{G}_*$ of Turing machines on conveyor belts.* Let us now define an automorphism θ that intuitively moves all (decorated) heads to the right, *disregarding the conveyor belt structure*, allowing heads to visit areas outside of their conveyor belts. This is completely *ad hoc*, and only used to build three specific automorphisms in §5.2.5.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

First, split $G = [\![0, 5]\!]$ into $G \simeq \{+1, -1\} \times [\![0, 2]\!]$, where $\{+1, -1\}$ is its *sign component*. Now every symbol of Σ contains a sign $\{+1, -1\}$, either in the sign-component of the state, or as its conveyor bit. Let $\pi_{\text{sign}} : \Sigma \to \{+1, -1\}$ return the sign of any symbol in Σ bit, that is,

$$\pi_{\text{sign}}(z) = \begin{cases} \varepsilon & \text{if } z = ((a, b), \varepsilon) \in \Gamma^2 \times \{+1, -1\}, \\ \varepsilon & \text{if } z = ((q, a), b) \in (Q \times \Gamma) \times \Gamma \text{ or } z = (a, (q, b)) \in (\Gamma \times (Q \times \Gamma)), \\ & \text{where } q = (q_0, d, x) \text{ and } x \simeq (\varepsilon, x') \in G. \end{cases}$$

We can then define the automorphism θ that moves every head to the right while leaving this sign symbol intact. More precisely, let $\sigma_{\text{sign}} : Q \times \{+1, -1\} \to Q$ rewrite the sign of the state:

$$\sigma_{\text{sign}} : ((q_0, d, (\_, x')),\ \varepsilon) \to (q_0, d, (\varepsilon, x')),$$

that is, $\pi_{\text{sign}}(\sigma_{\text{sign}}(q, \varepsilon)) = \varepsilon$, let $\pi_{\text{up}} : \Sigma \to \Gamma$ return the top tape-letter, $\pi_{\text{down}} : \Sigma \to \Gamma$ return the bottom tape-letter, and $\pi_Q : ((Q \times \Gamma) \times \Gamma) \cup (\Gamma \times (Q \times \Gamma)) \to Q$ return the state. Define $\theta : \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$ as

$$\theta(x)_i = \begin{cases} ((q', a), b) & \text{if } x_{i-1} \in (Q \times \Gamma) \times \Gamma, \\ (a, (q', b)) & \text{if } x_{i-1} \in \Gamma \times (Q \times \Gamma), \\ ((a, b), \pi_{\text{sign}}(x_i)) & \text{if } x_{i-1} \in \Gamma^2 \times \{+1, -1\}, \end{cases}$$

for $x \in \Sigma^{\mathbb{Z}}$, where $a = \pi_{\text{up}}(x_i)$, $b = \pi_{\text{down}}(x_i)$, and q' is the state of x_{i-1} with sign component $\pi_{\text{sign}}(x_i)$, that is, $q' = \sigma_{\text{sign}}(\pi_Q(x_{i-1}), \pi_{\text{sign}}(x_i))$.

Then θ is an automorphism of bi-radius 1, and we define by

$$\mathcal{G}_* = \langle \mathcal{G} \cup \{\theta\} \rangle$$

the finitely generated group generated by θ and the Turing machine instructions of $\mathcal{G}$. This finitely generated group $\mathcal{G}_*$ is where we prove distortion in Lemma 5.1.

5.1.5. *Main result: distortion in* $\Sigma^{\mathbb{Z}}$. Let $\mathcal{M}_0 = (Q_0, \Gamma_0, \Delta_0)$ be an arbitrary Turing machine. Denoting by $Q = Q_0 \times D \times G$ and $\Gamma = \Gamma_0$ the decorated versions of Q_0 and Γ_0, we define the symmetrized Turing machine $\mathcal{M} = (Q, \Gamma, \Delta)$ that acts forward in time on ducks $d_\rightarrow$ and backward on ducks $d_\leftarrow$:

$$\begin{aligned} \Delta = & \bigcup_{x \in G} \{((q, d_\rightarrow, x), a, (q', d_\rightarrow, x), b) : (q, a, q', b) \in \Delta_0\} \\ & \quad \cup \{((q, d_\rightarrow, x), \delta, (q', d_\rightarrow, x)) : (q, \delta, q') \in \Delta_0\} \\ & \cup \bigcup_{x \in G} \{((q', d_\leftarrow, x), b, (q, d_\leftarrow, x), a) : (q, a, q', b) \in \Delta_0\} \\ & \quad \cup \{((q', d_\leftarrow, x), \delta, (q, d_\leftarrow, x)) : (q, \delta, q') \in \Delta_0\}. \end{aligned}$$

We can now establish Lemma 5.1, which is the main result of this section.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

LEMMA 5.1. *Assume some Turing machine $\mathcal{M}_0$ satisfies the three properties of Lemma 4.2. Then, denoting by $\mathcal{M}$ its symmetrized (and decorated) version, the automorphism $(f_{\mathcal{M}})^n$ of $\mathrm{Aut}(\Sigma^{\mathbb{Z}})$ (that is, the action of $\mathcal{M}$ on the conveyor belts of $\Sigma^{\mathbb{Z}}$) has word norm $O(\log^{p+1} n + \log^2 n)$ in $\mathcal{G}_*$.*

Remark 5.2. Note that the assumption in this lemma focuses on the original machine $\mathcal{M}_0$ (and not $\mathcal{M}$) verifying the properties of Lemma 4.2, while the conclusion of this lemma focuses on the symmetrized version $(f_{\mathcal{M}})^n$ (and not $(f_{\mathcal{M}_0})^n$).

5.1.6. *Overview.* As the decorated SMART machine satisfies Lemma 4.2, combining it with Lemma 5.1 above, we obtain a distortion element of infinite order in $\mathcal{G}_*$, whose powers have word norm $O(\log^5 n)$. However, this does not yet prove Theorem A: to obtain an upper bound $O(\log^4 n)$ on the word norm, we add a few additional tricks and optimizations in §5.4.

This section focuses on the proof of Lemma 5.1 and Theorem A.

Intuitively, we prove Lemma 5.1 by applying the ℓ-cyclic automorphism $T_{\ell,\mathcal{M}}$ in the conveyor belts of length ℓ. The automorphisms $T_{\ell,\mathcal{M}}$ are finitarily distorted, and the distortion is transported from the cyclic automorphisms to $f_{\mathcal{M}}$.

Section 5.2 develops several tools for this proof. First, we build the tools to condition the application of automorphisms by the length of conveyor belts, as to apply the correct $T_{\ell,\mathcal{M}}$ to the conveyor belts of the correct length. Morally, $f_{\mathcal{M}}$ is then the infinite product of all the $T_{\ell,\mathcal{M}}$.

Then, we develop a method called the *two-scale trick* to express $f_{\mathcal{M}}$ as a product of finitely many $T_{\ell,\mathcal{M}}$. It generates temporary conveyor belts of sufficient length so as to move the head (with finitary distortion) without it 'seeing' the temporary borders, and then erases them.

Section 5.3 then contains the proof of Lemma 5.1. As mentioned above, §5.4 covers various optimizations in the case of the SMART machine, to obtain the final degree four of polylogarithmic norm growth.

Remark 5.3. At this point, the reader may think that conveyor belts are a restriction imposed by the context (as a way to embed Turing machines into a full shift); and that proving distortion in a similar setting without conveyor belts, for example, on the bi-infinite tapes of $\mathcal{X}$ (that is, in the groups of generalized Turing machines $\mathrm{RTM}(n, k)$) would be easier.

While the former is true, we think the latter is misleading. Indeed, the idea of temporary conveyor belts (from the two-scale trick) is a key component of our proof of distortion, even in the groups $\mathrm{RTM}(n, k)$. Without the ability to mark and erase temporary borders, we do not know how to prove the distortion of (what morally is) the SMART machine.

5.2. *Permutation engineering in $\Sigma^{\mathbb{Z}}$.* Similarly to §4.2.1, in this new setting, the states of Q once again have a ghost component G. We separate Q into two components: $Q = Q' \times G$ (so, with the notation $Q = Q_0 \times D \times G$, we have $Q' = Q_0 \times D$).

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

With similar notation, the set of heads $H = Q \times \Gamma$ also splits as $H = H' \times G$, where $H' = Q' \times \Gamma$. Note that for any $g \in \mathrm{Sym}(H')$, the permutation $g \times \mathrm{id}_G$ belongs to $\mathrm{Sym}(H)$.

In this section, we focus on building a new sort of gate-conditioning: conditioning on the structure of conveyor belts. To proceed, we somewhat follow the same ideas that we already developed in §4.2.1. Finally, we prove that we can modify the conveyor belt structure in §5.2.5.

5.2.1. *Structure conditioning.* Let us define by $\mathcal{CB}$ the set of conditions that focus on the structure of conveyor belts.

Given a head and a conveyor belt, we denote by $\mathrm{len} \in \mathbb{N}$ the length of the conveyor belt (that is, the length of the underlying finite cyclic tape). The integers $l_{\mathrm{r}} \in \mathbb{N}$ and $l_{\mathrm{l}} \in \mathbb{N}$ refer to the distance between a head and respectively the right and left border of the conveyor belt. In this context, $\mathcal{CB}$ is defined as the smallest set of conditions satisfying the following.

- For every $n \in \mathbb{N}$, the condition $\mathrm{len}|n$ (that is, the length of the conveyor belt divides n) belongs to $\mathcal{CB}$.
- The conditions $l_{\mathrm{r}} = 0$ and $l_{\mathrm{l}} = 0$ belong to $\mathcal{CB}$.
- For $n \in \mathbb{Z}$, the condition $\rho^n(C)$ belongs to $\mathcal{CB}$.
- If $C_1, C_2 \in \mathcal{CB}$, then $C_1 \wedge C_2 \in \mathcal{CB}$.
- If $C_1, C_2 \in \mathcal{CB}$, then $C_1 \vee C_2 \in \mathcal{CB}$.
- If $C \in \mathcal{CB}$, then $\neg C \in \mathcal{CB}$.

Here, for C a condition, the condition $\rho^n(C)$ (for $n \in \mathbb{Z}$) holds if and only if C holds after applying ρ^n.

For any permutation $g \in \mathrm{Sym}(H)$ and any condition $C \in \mathcal{CB}$, define $f_{g,C}$ to be the conditioned gate that applies g on a head if and only if this head belongs to a conveyor belts that verifies the condition C.

LEMMA 5.4. *For any permutation $g \in \mathrm{Sym}(H')$ and condition $C \in \mathcal{CB}$, the conditioned gate $f_{g\times\mathrm{id},C}$ belongs to $\mathcal{G}$. Additionally, if $T : \mathcal{CB} \to \mathbb{N}$ denotes the following function:*

$$T(C) = \begin{cases} n & \text{if } C \text{ is } \mathrm{len}|n, \\ 1 & \text{if } C \text{ is } l_{\mathrm{r}} = 0 \text{ or } l_{\mathrm{l}} = 0, \\ T(C_1) + |n| & \text{if } C = \rho^n(C_1), \\ 2T(C_1) + 2T(C_2) & \text{if } C = C_1 \wedge C_2, \\ 2T(C_1) + 2T(C_2) + 5 & \text{if } C = C_1 \vee C_2, \\ T(C_1) + 1 & \text{if } C = \neg C_1, \end{cases}$$

then the word norm of $f_{g\times\mathrm{id},C}$ in $\mathcal{G}$ is $O(T(C))$.

We divide the proof of this lemma into the following claims.

CLAIM 5.5. *Let $g \in \mathrm{Alt}(G)$ and $\ell \in \mathbb{N}$. Then the conditioned gate $f_{\mathrm{id}\times g,\mathrm{len}|\ell}$ (that applies $\mathrm{id}_{H'} \times g$ on conveyor belts whose length divides ℓ) belongs to $\mathcal{G}$ with word norm $O(\ell)$.*

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Proof. Let $r' = (0, 1, \dots, |\Gamma| - 1) \in \mathrm{Sym}(\Gamma)$ and $r = \mathrm{id}_Q \times r' \in \mathrm{Sym}(H)$ (recall $H = Q \times \Gamma$) be the rotation of the tape-letter under the head. This proof relies on the following trivial observation: applying f_r changes the letter at both position 0 and position ℓ if and only if the length of the conveyor belt divides ℓ (since the tapes are cyclic).

Let $C = \bigcup_{a \in \Gamma}([Q \times \{a\}]_0 \cap [a]_\ell)$ be the condition that checks whether cells at position 0 and ℓ have the same tape-letter, and $g \in \mathrm{Alt}(G)$. Since G has cardinality at least five, by Ore's theorem [**42**, Theorem 7], there exist $g_1, g_2 \in \mathrm{Alt}(G)$ such that $g = [g_1, g_2]$; and according to Lemma 4.6, both $f_{\mathrm{id} \times g_1, C}$ and $f_{\mathrm{id} \times g_2, C}$ belong to $\mathcal{G}$ with word norm $O(\ell)$. Then,

$$f_{\mathrm{id} \times g, \mathrm{len}|\ell} = [f_{\mathrm{id} \times g_1, C},\ f_r \circ f_{\mathrm{id} \times g_2, C} \circ f_r^{-1}].$$ □

CLAIM 5.6. *Let $g \in \mathrm{Alt}(G)$ and C be a condition $l_\mathrm{r} = 0$ or $l_\mathrm{l} = 0$. Then the conditioned gate $f_{\mathrm{id} \times g, C}$ (that applies $\mathrm{id}_{H'} \times g$ on heads that are respectively on the left or right border of their conveyor belt) belongs to $\mathcal{G}$ with word norm $O(\ell)$.*

Proof. As $|G| \geq 5$, by Ore's theorem, there exists $g_1, g_2 \in \mathrm{Alt}(G)$ such that $g = [g_1, g_2]$. Applying the commutator trick,

$$[f_{g_1}^{\mathrm{up}},\ \rho^{-1} \circ f_{g_2}^{\mathrm{down}} \circ \rho]$$

applies $f_{[g_1, g_2]}$ on heads that are exactly in the top-right corner of their conveyor belts. Similar formulas exist for bottom-right, top-left, and bottom-left corners of conveyor belts, so that one can condition any f_g on being applied on heads that are in the left or right corners of their conveyor belts with word norm $O(1)$. □

We can now proceed with the proof of Lemma 5.4. This proof is very similar to the proofs of Lemmas 4.6 and 4.7.

Proof of Lemma 5.4. By the proof of Lemma 4.7, we only need to prove that for every $g \in \mathrm{Alt}(G)$ and $C \in \mathcal{CB}$, the conditioned gate $f_{\mathrm{id} \times g, C}$ belongs to $\mathcal{G}$.

Let $C \in \mathcal{CB}$. If C is $\mathrm{len}|n$, then for any $g \in \mathrm{Alt}(G)$, the conditioned gate $f_{g,C}$ belongs to $\mathcal{G}$ with word norm $O(n)$ by Claim 5.5. If C is $l_\mathrm{r} = 0$ or $l_\mathrm{l} = 0$, then $f_{g,C}$ belongs to $\mathcal{G}$ with word norm $O(1)$ by Claim 5.6. If $C = \rho^n(C_1)$ for some $C_1 \in \mathcal{CB}$, then for any $g \in \mathrm{Alt}(G)$, we have $f_{g,C} = \rho^{-n} \circ f_{g,C_1} \circ \rho^n$.

Finally, if $C = C_1 \wedge C_2$, $C = C_1 \vee C_2$, or $C = \neg C_1$, the proof of Lemma 4.6 applies *mutatis mutandis*. □

Remark 5.7. The very same proof shows that $f_{g \times \mathrm{id}}^{\mathrm{up}}$ and $f_{g \times \mathrm{id}}^{\mathrm{down}}$ can also be conditioned by $\mathcal{CB}$.

5.2.2. *Corollary: conditioning on the length of conveyor belts.* For $\ell \in \mathbb{N}$ and $\sim \in \{<, \leq, =, \geq, >\}$, let us define conditions $\mathrm{len} \sim \ell$ (comparisons on the length of the conveyor belt).

LEMMA 5.8. *Let $\ell \in \mathbb{N}$ and $\sim \in \{<, \leq, =, \geq, >\}$. If $C = \mathrm{len} \sim \ell$, then C belongs to $\mathcal{CB}$ and $T(C) = O(\ell^2)$.*

In other words, for any $g \in \mathrm{Sym}(H')$, the conditioned gates $f_{g \times \mathrm{id}, C}$ belong to $\mathcal{G}$ with word norm $O(\ell^2)$.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Proof. By the proof of Lemma 4.7, we only need to prove that for every $g \in \mathrm{Alt}(G)$ and condition $C = \mathrm{len} \sim \ell$, the conditioned gate $f_{\mathrm{id}\times g, C}$ belongs to $\mathcal{G}$.

Assume that $\sim$ is equality. By going through the divisors of ℓ in decreasing order, we can build any $f_{\mathrm{id}\times g,\mathrm{len}=\ell}$ with word norm $O(\ell^2)$.(The number of divisors function d satisfies $d(\ell) = o(\ell^\epsilon)$ for any $\epsilon > 0$, so we even get word norm $O(\ell^{1+\epsilon})$ for $f_{\mathrm{id}\times g,\mathrm{len}=\ell}$.) For example, if $\ell = 6$,

$$f_{\mathrm{id}\times g,\mathrm{len}=6} = f_{\mathrm{id}\times g,\mathrm{len}|1} \circ f_{\mathrm{id}\times g^{-1},\mathrm{len}|2} \circ f_{\mathrm{id}\times g^{-1},\mathrm{len}|3} \circ f_{\mathrm{id}\times g,\mathrm{len}|6}.$$

(Our conveyor belts cannot actually have length 1, so $f_{\mathrm{id}\times g,\mathrm{len}|1}$ may be dropped.)

If $\sim$ is $\leq$, we similarly build $f_{\mathrm{id}\times g,\mathrm{len}\leq\ell}$ with word norm $O(\ell^2)$ by going through the interval $[\![1, \ell]\!]$ in decreasing order and picking suitable powers of g. For example, if $\ell = 6$,

$$f_{\mathrm{id}\times g,\mathrm{len}\leq 6} = f_{\mathrm{id}\times g^{-2},\mathrm{len}|1} \circ f_{\mathrm{id}\times g^{-1},\mathrm{len}|2} \circ f_{\mathrm{id}\times g^{0},\mathrm{len}|3} \circ f_{\mathrm{id}\times g,\mathrm{len}|4} \circ f_{\mathrm{id}\times g,\mathrm{len}|5} \circ f_{\mathrm{id}\times g,\mathrm{len}|6}.$$

So we obtain $f_{\mathrm{id}\times g,\mathrm{len}\sim\ell}$ for the relations $\sim \ell$ with $\sim \in \{\leq, =\}$. From this, the automorphisms with $\sim \in \{<, \geq, >\}$ are easy to obtain with elementary boolean algebra: for example, $\mathrm{len} > \ell$ is the condition $\neg(\mathrm{len} \leq \ell)$. □

Remark 5.9. One may view the above proof as an instance of Möbius inversion. If g has order m, take $K = \oplus_{\mathbb{Z}_+} \mathbb{Z}_m$ the commutative ring of infinitely many copies of $\mathbb{Z}_m$. We see K as keeping track of how many times g is applied at each conveyor belt length. Define functions $\iota, \gamma : \mathbb{Z}_+ \to K$ where $\iota(n)$ as the indicator function of n (as an element of K), and $\gamma(n)$ the indicator function of the divisor poset of n. Then $\gamma(n) = \sum_{d|n} \iota(d)$ so by Möbius inversion, $\iota(n) = \sum_{d|n} \gamma(d)\mu(d, n)$, where μ is the Möbius function of the divisibility poset; thus $\mu(d, n)$ tells us which power of g we should use for each divisor to get $\iota(n)$. The values of ι are a basis of K, so we can get other conditional applications of g with linear combinations.

5.2.3. *Corollary: conditioning on the distance to borders.* Recall that $l_\mathrm{r} \in \mathbb{N}$ and $l_\mathrm{l} \in \mathbb{N}$ denote the distance between the head and the right (respectively left) border of the conveyor belt.

LEMMA 5.10. *For $t \in \mathbb{N}$ and $\sim \in \{\leq, <, =, >, \geq\}$, let C be a condition of the form $l_\mathrm{r} \sim t$ (respectively $l_\mathrm{l} \sim t$).*

Then C belongs to $\mathcal{CB}$ and $T(C) = O(t^2)$. In other words, for any $g \in \mathrm{Sym}(H')$, the conditioned gate $f_{g\times\mathrm{id},C}$ belongs to $\mathcal{G}$ with word norm $O(t^2)$.

Proof. Very similarly to the proof of Lemma 4.13, we rely on a divide and conquer approach. For example,

$$\begin{aligned}
&l_\mathrm{r} \leq t \text{ is equivalent to } \rho^{-t/2}(l_\mathrm{r} \leq t/2) \ \vee \ \rho^{t/2}(l_\mathrm{r} \leq t/2),\\
&l_\mathrm{r} = t \text{ is equivalent to } (l_\mathrm{r} \leq t) \ \wedge \ \neg(l_\mathrm{r} \leq t-1),\\
&l_\mathrm{r} < t \text{ is equivalent to } l_\mathrm{r} \leq t-1.
\end{aligned}$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

The rest of the comparisons on l_{r} follow using basic Boolean algebra; and the case of l_{l} is symmetric. □

5.2.4. *Corollary: from cyclic tapes C_ℓ to conveyor belts in $\Sigma^{\mathbb{Z}}$.* For $T \in \mathcal{G}_\ell$, there exists by definition some $T_1, \dots, T_N \in \{\pi_g \mid g \in \mathrm{Sym}(Q \times \Gamma)\} \cup \{\rho_q \mid q \in Q\}$ such that $T = T_N \circ \cdots \circ T_1$. Then, as each generator π_g with $g \in \mathrm{Sym}(Q \times \Gamma)$ (respectively ρ_q of $\mathcal{G}_\ell$) corresponds to a generator f_g of $\mathcal{G}$ (respectively ρ_q of $\mathcal{G}$), T defines an automorphism f_T of $\mathrm{Aut}(\Sigma^{\mathbb{Z}})$ of the same word norm. By construction, for any $T \in \mathcal{G}_\ell$, the corresponding $f_T \in \mathcal{G}$ acts like T on conveyor belts of length ℓ (and produces garbage on conveyor belts of length $\neq \ell$).

Remark 5.11. The choice of f_T is not canonical, in the sense that two different presentations $T_1, \dots, T_N$ and $T'_1, \dots, T'_{N'}$ of T may define two different automorphisms f_T (they would define the same action on conveyor belts of length ℓ, but produce different garbage on conveyor belts of length $\neq \ell$). This has no importance in what follows, but for the sake of cleanliness, fix an arbitrary order on the generators of $\mathcal{G}_\ell$ and define f_T from the lexicographically minimal presentation (among the presentations of shortest length) of T.

We now use the previous lemma to condition f_T so that it acts only in conveyor belts of length ℓ. Recall that $T/_{d_\rightarrow}$ acts like T on ducks $d = d_\rightarrow$, and is the identity otherwise, and denote T_s the symmetrized version of T, that is

$$T_s(w) = \begin{cases} T(w) & \text{if } w \in C_\ell[d_\rightarrow], \\ T^{-1}(w) & \text{if } w \in C_\ell[d_\leftarrow], \\ w & \text{otherwise.} \end{cases}$$

LEMMA 5.12. *Assume $T/_{d_\rightarrow}$ belongs to $\mathcal{G}_\ell$. Then $f_{T_s,\mathrm{len}=\ell} \in \mathrm{Aut}(\Sigma^{\mathbb{Z}})$ belongs to $\mathcal{G}$ with word norm $O(\|T/_{d_\rightarrow}\| + \ell^2)$.*

Proof. Let $d \in \mathrm{Sym}(Q \times \Gamma)$ be the involution that swaps $d_\rightarrow$ and $d_\leftarrow$ on the head. By Lemma 5.8, $f_{d,\mathrm{len}=\ell}$ has word norm $O(\ell^2)$ and

$$f_{T_s,\mathrm{len}=\ell} = f_{d,\mathrm{len}=\ell} \circ (f_{T/_{d_\rightarrow}})^{-1} \circ f_{d,\mathrm{len}=\ell} \circ (f_{T/_{d_\rightarrow}}).$$ □

Remark 5.13. Note that the assumption of this lemma focuses on $T/_{d_\rightarrow}$, but the conclusion focuses on T_s. Note also that while f_T is not canonical (see Remark 5.11), $f_{T_s,\mathrm{len}=\ell}$ is.

5.2.5. *Creating/erasing conveyor belts.* Let $\tau_{\mathrm{cb}} = \mathrm{id}_{H'} \times \tau' \in \mathrm{Sym}(H)$, where $\tau' \in \mathrm{Sym}(G)$ is the involution that permutes the sign $+1 \leftrightarrow -1$ in G (when considering G as $G \simeq \{+1, -1\} \times [\![0, 2]\!]$).

For $t \in \mathbb{N}$, define

$$f_{\tau_{\mathrm{cb}},\rightarrow t} = \theta^{-t} \circ f_{\tau_{\mathrm{cb}}} \circ \theta^t,$$
$$f_{\tau_{\mathrm{cb}},t\leftarrow} = \theta^{t} \circ f_{\tau_{\mathrm{cb}}} \circ \theta^{-t},$$
$$f_{\tau_{\mathrm{cb}},t\leftrightarrow t} = f_{\tau_{\mathrm{cb}},\rightarrow t} \circ f_{\tau_{\mathrm{cb}},t\leftarrow},$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

where $\theta \in \mathcal{G}_*$ (defined in §5.1.4) is the right movement of heads ignoring the conveyor belt structure. These automorphisms all belong to $\mathcal{G}_*$ with word norm $O(t)$, and they modify the conveyor belt structure: they can create or erase conveyor belts.

In general, these modifications to the conveyor belt structure are quite unpredictable (and meaningless), so we will only apply them through conjugation (that is, by conjugating automorphisms with 'small support' by these automorphisms), so that they effectively only act in very specific situations.

5.3. *Proof of Lemma 5.1.* Let us briefly recall the statement of Lemma 5.1. For the precise context and notation, see §5.1.5.

LEMMA 5.1. *Assume some Turing machine $\mathcal{M}_0$ satisfies the three properties of Lemma 4.2. Then, denoting by $\mathcal{M}$ its symmetrized (and decorated) version, the automorphism $(f_{\mathcal{M}})^n$ of $\mathrm{Aut}(\Sigma^{\mathbb{Z}})$ (that is, the action of $\mathcal{M}$ on the conveyor belts of $\Sigma^{\mathbb{Z}}$) has word norm $O(\log^{p+1} n + \log^2 n)$ in $\mathcal{G}_*$.*

Before going into the precise math, here is an overview of the proof. The main idea consists in building $(f_{\mathcal{M}})^n$ (the action of $\mathcal{M}$ on the conveyor belts of $\Sigma^{\mathbb{Z}}$) from the automorphisms $(T_{\ell,\mathcal{M}_0})^n$ (each action of $\mathcal{M}_0$ on each length ℓ of finite cyclic tapes C_ℓ), the latter being finitarily distorted by hypothesis (that is, having small word norm $O(\ell^p)$). Conditioning each $(T_{\ell,\mathcal{M}_0})^n$ on the correct length of conveyor belt with Lemma 5.12, we can use these automorphisms and the distortion carries from finite cyclic tapes to conveyor belts.

Informally, $(f_{\mathcal{M}})^n$ then becomes the product of infinitely many $(T_{\ell,\mathcal{M}})^n$. To write $(f_{\mathcal{M}})^n$ as a finite product, consider the movement of $\mathcal{M}$ and notice that $m(n)$—the number of cells visited by $\mathcal{M}$ in n steps—is assumed to be $O(\log n)$. As a consequence, a head at distance more than $m(n)$ from the borders of its conveyor belts will not see the borders in question. In such a case, we can create temporary conveyor belts of size $O(\log n)$, apply the corresponding $(T_{\ell,\mathcal{M}})^n$, and erase the temporary borders: the applied operation coincides with $(f_{\mathcal{M}})^n$ in the original large conveyor belt.

Let us formalize these ideas. In particular, generating temporary conveyor belts is not a reversible operation: to solve this issue, we introduce the *two-scale trick* (which introduces not one, but two temporary conveyor belts of different sizes, hence its name).

Proof of Lemma 5.1. Fix an integer, which is the power of $f_{\mathcal{M}}$ that we want to build. Without any loss of generality, we assume that it is even. Indeed, if some $2n+1$ is odd, then $2n$ is even, and $\|(f_{\mathcal{M}})^{2n+1}\| = \|(f_{\mathcal{M}})^{2n}\| + O(1)$. We denote this even integer by $2n$. With notation from Lemma 4.2, let $L = C \cdot \log n + C'$. By hypothesis, every $(T_{\ell,\mathcal{M}_0})^n$ has word norm $O(\ell^p)$ in $\mathcal{G}_\ell$.

First, we use Lemma 5.12 to condition each $(T_{\ell,\mathcal{M}_0})^{2n}$ to apply only on conveyor belts of length ℓ. In other words, since $\mathcal{M}$ is the symmetrized version of $\mathcal{M}_0$, we obtain that every automorphism

$$(f_{(T_{\ell,\mathcal{M}}),\mathrm{len}=\ell})^{2n} = f_{(T_{\ell,\mathcal{M}})^{2n},\mathrm{len}=\ell}$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

belongs to $\mathcal{G}$ with word norm $O(\ell^p)$, so that we can manage all conveyor belts of length $< 12L$ with word norm $O(L \cdot L^p)$:

$$(f_{\mathcal{M},\text{len}<12L})^{2n} = \prod_{\ell=1}^{6L-1} (f_{(T_{2\ell,\mathcal{M}}),\text{len}=2\ell})^{2n}$$

(recall that conveyor belts in $\Sigma^{\mathbb{Z}}$ are cyclic tapes of even length).

Then, to manage larger conveyor belts, we use what we call the *two-scale trick*. In the introductory paragraphs, we mentioned the idea of introducing temporary conveyor belts to move heads that originally belonged in very large conveyor belts, and then removing the temporary borders. The difficulty lies in properly removing the temporary conveyor belts once the machine has been applied. To solve this, we will actually use temporary conveyor belts twice, with different sizes. We give a visual explanation of this trick in Figure 7.

Define $L_1 = 4L - 2$, $L_2 = 8L - 2$, $L_3 = 12L - 2$. Note that L_1 (respectively L_2) is the length of a conveyor belt constructed by $f_{\tau_{\text{cb}},L\leftrightarrow L}$ (respectivelyt $f_{\tau_{\text{cb}},2L\leftrightarrow 2L}$) defined in §5.2.5. Then, let

$$\begin{aligned}\lambda_n = (&f_{\tau_{\text{cb}},2L\leftrightarrow 2L} \circ (f_{(T_{L_2,\mathcal{M}})^n,\text{len}=L_2}) \circ f_{\tau_{\text{cb}},L\leftrightarrow L} \circ (f_{(T_{L_1,\mathcal{M}})^n,\text{len}=L_1})^{-1} \\ &\circ f_{\tau_{\text{cb}},2L\leftrightarrow 2L} \circ (f_{(T_{L_1,\mathcal{M}})^n,\text{len}=L_1}) \circ f_{\tau_{\text{cb}},L\leftrightarrow L}).\end{aligned}$$

Let f_i denote the composition of the first i automorphisms on this list, that is, $f_1 = f_{\text{cb},2L\leftrightarrow 2L}, \ldots, f_7 = \lambda_n$. The actions of the inverses of the automorphisms f_i are illustrated in Figure 7 (on a certain subset of configurations).

Then, denote by $d \in \text{Sym}(H)$ the ducking involution, that is, the involution that flips ducks $d_{\rightarrow}$ and $d_{\leftarrow}$ in $Q = Q_0 \times D \times G$. Let $C \in \mathcal{CB}$ be the structural condition $(\text{len} \geq 12L) \wedge (l_{\text{l}} \geq 3L) \wedge (l_{\text{r}} \geq 3L)$ (see §5.2.1). By Lemmas 5.8, 5.10, and 5.4, the automorphism $f_{d,C}$ belongs to $\mathcal{G}$ with word norm $O(L^2)$. Defining

$$f_{\mathcal{M},2n,C} = (\lambda_n)^{-1} \circ (f_{d,C}) \circ (\lambda_n),$$

we see from Figure 7, reading the successive partial conjugations ${f_{d,C}}^{f_i}$ top-down, that $f_{d,C}$ gets conjugated to a map that applies the machine $(\mathcal{M})^n$ twice if it is on a conveyor belt of length $\geq 12L$ and the head is sufficiently far (that is, at distance at least $3L$) from both its left and right borders; and flips the duck as a side product.

Denoting by C_{r} the condition $(\text{len} \geq 12L) \wedge (l_{\text{l}} \geq 3L) \wedge (l_{\text{r}} < 3L)$ and by C_{l} the condition $(\text{len} \geq 12L) \wedge (l_{\text{l}} < 3L) \wedge (l_{\text{r}} \geq 3L)$, we can build automorphisms $f_{\mathcal{M},2n,C_{\text{r}}}$ (respectively $f_{\mathcal{M},2n,C_{\text{l}}}$) that manage heads in conveyor belts of size $\geq 12L$ containing heads at distance less than $3L$ from their right (respectively left) border.

By a very similar reasoning, $f_{\mathcal{M},2n,C_{\text{r}}}$, and $f_{\mathcal{M},2n,C_{\text{l}}}$ belong to $\mathcal{G}_*$ with word norm $O(L^{p+1} + L^2)$. We need to alter this reasoning only twice: first, instead of using $f_{\tau_{\text{cb}},\ell\leftrightarrow\ell}$ to create/erase borders both left and right, we use respectively $f_{\tau_{\text{cb}},\ell\leftarrow}$ and $f_{\tau_{\text{cb}},\rightarrow\ell}$ to create/erase borders only in one direction; second, as the size of the conveyor belt is no

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

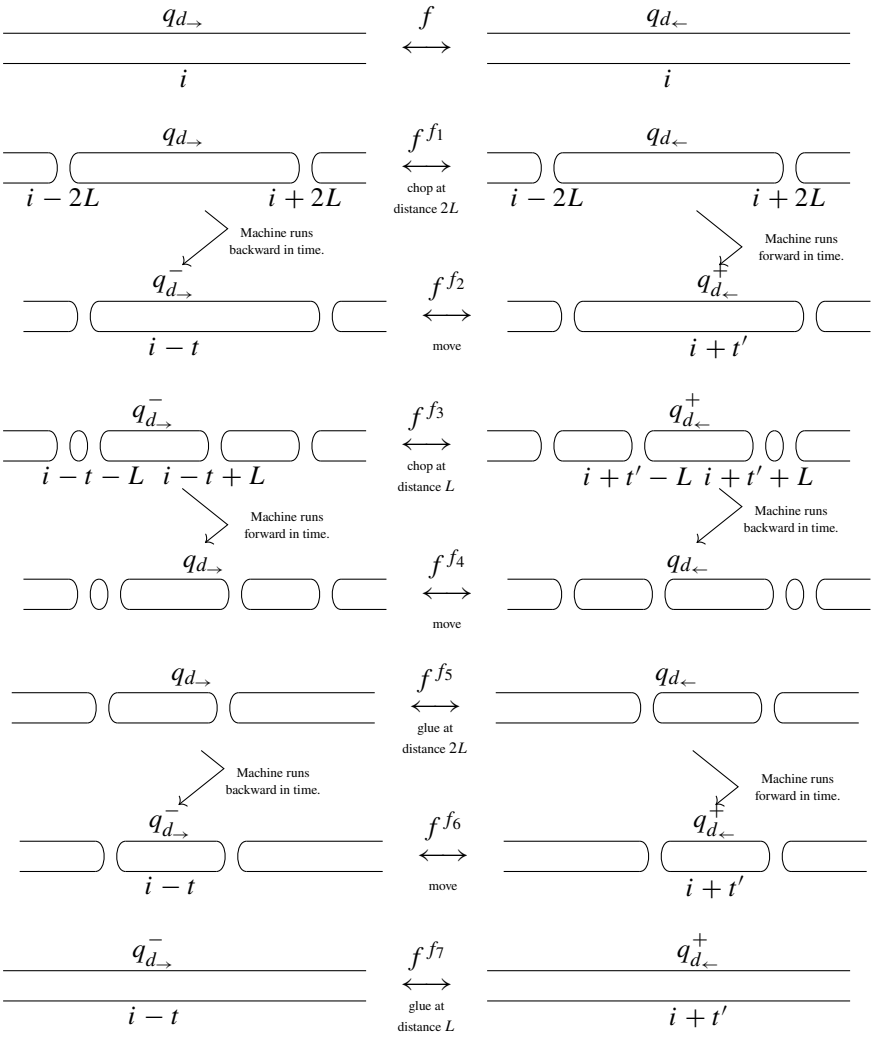


FIGURE 7. The 'two-scale trick'. Illustration of the two-scale trick. We show the conveyor belts as lines (without tape letters). The head may be on either track. The head is in position i in state q initially, was in position $i-t$ in state q^- at time $-n$, and will be in position $i+t'$ in state q^+ at time $+n$. The automorphism f acts trivially unless we are in the situation of the first line, so the conjugated automorphisms also act non-trivially only in the shown situation. In particular, f^{f_7} behaves as expected.

longer precisely L_1 or L_2, but only bounded by L_1 or L_2, we have to replace occurrences of $(f_{(T_{\ell,\mathcal{M}})^n,\text{len}=\ell})$ (for $\ell = L_1$ or $\ell = L_2$) in the previous formulas with

$$\prod_{j=1}^{\ell} f_{(T_{j,\mathcal{M}})^n,\text{len}=j}.$$

In any case, these permutations have disjoint support (because the distance to a conveyor belt border is checked 'after n steps of computation' in the conjugations ${\lambda_n}^{-1} \circ (f_{d,C}) \circ \lambda_n$) and word norm $O(L^{p+1} + L^2)$. We obtain that

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

$$(f_{\mathcal{M}})^{2n} = (f_{d,\text{len} \geq 12L} \circ f_{\mathcal{M},2n,C_r} \circ f_{\mathcal{M},2n,C_l} \circ f_{\mathcal{M},2n,C}) \circ \left(\prod_{n=1}^{6L-1} (f_{T_{\ell,\mathcal{M}},\text{len}=2l})^{2n} \right),$$

which concludes the proof. □

5.4. *Improving the upper bound on SMART.* Lemma 5.1 was a general result telling how finitely distorted machines give rise to distorted automorphisms on full shifts. In the context of the SMART machine, we provide some additional improvements to get the upper bound from $O(\log^5 n)$ to $O(\log^4 n)$.

These improvements are based on the following idea: the computation of the $(T_{\ell,\mathcal{S}})^n$ is 'uniform' in ℓ, as the kth step of the encoding (from Lemma 4.3) and the kth step of the addition (from Lemma 4.4) are the same on all conveyor belts of length $\geq k$. Thus, we can compute the action of SMART on all conveyor belts in parallel, improving the word norm as mentioned. We also make minor optimizations to the alphabet, by joining the decorations used for the decorated SMART and the decorated automorphism.

All in all, Lemma 5.14 concludes the proof of Theorem A. It also provides a self-contained result about distortion.

LEMMA 5.14. *Let $\mathcal{S} = (\Gamma^{(o)}, Q^{(o)}, \Delta^{(o)})$ be the original SMART machine introduced in §3, and define its decorated symmetrized version $\mathcal{S} = (\Gamma, Q, \Delta)$ as follows:*

$$\begin{aligned}
\Gamma &= \Gamma^{(o)},\\
Q &= Q^{(o)} \times D \times G,\\
\Delta &= \bigcup_{x \in G} \{((q, d_\rightarrow, x), a, (q', d_\rightarrow, x), b) : (q, a, q', b) \in \Delta^{(o)}\}\\
&\qquad \cup \{((q, d_\rightarrow, x), \delta, (q', d_\rightarrow, x)) : (q, \delta, q') \in \Delta^{(o)}\}\\
&\quad \cup \bigcup_{x \in G} \{((q', d_\leftarrow, x), b, (q, d_\leftarrow, x), a) : (q, a, q', b) \in \Delta^{(o)}\}\\
&\qquad \cup \{((q', d_\leftarrow, x), \delta, (q, d_\leftarrow, x)) : (q, \delta, q') \in \Delta^{(o)}\},
\end{aligned}$$

where $D = \{d_\rightarrow, d_\leftarrow\}$ and $G = [\![0, 5]\!] \simeq \{+1, -1\} \times [\![0, 2]\!]$; and denote

$$\Sigma = (\Gamma^2 \times \{+1, -1\}) \sqcup ((Q \times \Gamma) \times \Gamma) \sqcup (\Gamma \times (Q \times \Gamma)).$$

Then $(f_{\mathcal{S}})^n$ has word norm $O(\log^4 n)$ in a finitely generated subgroup $\mathcal{G}_*$ of $\text{Aut}(\Sigma^{\mathbb{Z}})$.

Proof. Once again, we prove that $(f_{\mathcal{S}})^n$ has word norm $O(\log^4 n)$ in the following subgroup of $\text{Aut}(\Sigma^{\mathbb{Z}})$:

$$\begin{aligned}
\mathcal{G}_* = \langle \{ &f_g^{\text{up}}, f_g^{\text{down}}, f_g : g \in \text{Sym}(Q \times \Gamma)\}\\
&\cup \{\rho_q \mid q \in Q\} \cup \{\theta\}\rangle.
\end{aligned}$$

We start by explaining the new alphabet. We have dropped the duck $\{d_1, d_2\}$ to fuse it with $\{d_\rightarrow, d_\leftarrow\}$. Our assumption in the previous section is that we can efficiently apply the Turing machine when the duck is $d_\rightarrow$, while doing nothing on ducks $d_\leftarrow$, and this is exactly how the duck was used in §4 (see Lemma 4.5).

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

We also have fused the two ghosts: as both ghosts were only used temporarily in our construction, always returning to their original value after each step of computation, it is safe to fuse them.

Now we explain the optimization; we only give a high-level explanation, as this is completely analogous to what was done in the previous section. We only amend the proof above by proving that both $(f_{\mathcal{M},\mathrm{len}<L_3})^{2n}$ and $(f_{\mathcal{M},\mathrm{len}\le L_3})^{2n}$ have word norm $O(L^4)$, as they are sufficient to manage both small conveyor belts of length $< L_3$ and large conveyor belts in the two-scale trick.

(1) The encoding (word norm $O(L^4)$). Let $\tilde{F}$ be either $\tilde{F}_{\mathrm{init}}$, $\tilde{F}_{k\to k+1}$, or $\tilde{F}_{\ell,\mathrm{final}}$, that is, one of the steps of encoding of $\mathcal{S}$ defined in Lemma 4.20. Recall that

$$\tilde{F}(w) = \begin{cases} F(w) & \text{if } w \in C_\ell[d_\rightarrow], \\ F^{-1}(w) & \text{if } w \in C_\ell[d_\leftarrow], \\ w & \text{otherwise.} \end{cases}$$

As explained in §5.2.4, the generators of $\mathcal{G}_\ell$ correspond to generators of $\mathcal{G}_*$, so that $\tilde{F}$ can also be considered as an element $f_{\tilde{F}}$ of $\mathcal{G}_*$, which acts like $\tilde{F}$ on conveyor belts of length ℓ, and produces garbage on conveyor belts of length $\neq \ell$.

The key point of this proof consists in noticing that $\tilde{F}_{k\to k+1}$ behaves correctly on every conveyor belt of length $\ge k+2$, and produces garbage on conveyor belts of length $\le k+1$.

Recall that each $\tilde{F}$ was a piecewise-defined bijection defined as a product of finitely many $\tilde{F}_{U,V}$. Each $\tilde{F}_{U,V}$ was built in Lemma 4.18 by conjugating some conditioned gate $\pi_{d,U}$ ($d \in \mathrm{Sym}(H)$ is the ducking involution, which swaps ducks $d_\rightarrow$ and $d_\leftarrow$) with some automorphism $g \in \mathcal{G}_\ell$.

Combining Lemmas 4.7 and 5.2.1, we now condition simultaneously on the content of the tape (that is, the condition U in the proof of Lemma 4.20) and the structure of conveyor belts. In other words, for any structural condition $C \in \mathcal{CB}$, the automorphism $f_{d,U\wedge C}$ belongs to $\mathcal{G}_*$ and we can define

$$f_{\tilde{F}_{U,V},C} = f_{g^{-1}} \circ f_{d,U\wedge C} \circ f_g$$

(where $f_g \in \mathcal{G}_*$ denotes the automorphism $g \in \mathcal{G}_\ell$ considered as an element of $\mathcal{G}_*$).

Then $f_{\tilde{F}_{U,V},C}$ acts like $\tilde{F}_{U,V}$ on conveyor belts which respect the structural condition C. And since the word norm of $\tilde{F}_{U,V}$ was already the word norm of $g \in \mathcal{G}_\ell$, that is, $O(\ell^3)$, we obtain that $f_{\tilde{F}_{U,V},C}$ also has word norm $O(\ell^3 + T(C))$. Composing finitely many of those, and adding a final $f_{d,C}$, we obtain automorphisms $f_{\tilde{F},C}$ for $\tilde{F}$ being either $\tilde{F}_{\mathrm{init}}$, $\tilde{F}_{k\to k+1}$, or $\tilde{F}_{\ell,\mathrm{final}}$.

Now, we consider the following automorphisms (for $0 \le k < L_3 - 2$):

$$f_{\tilde{F}_{\mathrm{init}},\mathrm{len}<L_3}, \quad f_{\tilde{F}_{k\to k+1},k+2\le\mathrm{len}<L_3}, \quad f_{\tilde{F}_{\ell,\mathrm{final}},\mathrm{len}=\ell}.$$

Each of these elements has word norm $O(L_3^2 + L_3^3)$. Then, define

$$f_{\mathrm{encode}} = \left(\prod_{\ell=1}^{L_3-1} f_{\tilde{F}_{\ell,\mathrm{final}},\mathrm{len}=\ell}\right) \circ \left(\prod_{k=1}^{L_3-3} f_{\tilde{F}_{k\to k+1},k+2\le\mathrm{len}<L_3}\right) \circ f_{\tilde{F}_{\mathrm{init}},\mathrm{len}<L_3}.$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

The automorphism f_{encode}, which acts on conveyor belts of length $< L_3$ and encodes SMART configurations with ducks $d_\rightarrow$ into their correct encoding, and produces garbage on ducks $d_\leftarrow$, has word norm $O(L^4)$. (A similar automorphism obviously exists for conveyor belts of length $\leq L_3$.)

(2) The addition of $2n$ on ducks $d_\rightarrow$ (word norm $O(L^3)$). We follow the proof of Lemma 4.4, which applied the 'school algorithm' for additions. The key idea of this proof is that the jth digit of $2n$ can be applied with the same automorphism to every conveyor belt of length $\geq j$.

As in the proof of Lemma 4.4, we first manage the addition of $2n$ modulo $2 \cdot 3^\ell$ in every conveyor belt of length $\ell < L_3 = 12L - 2$ with the school-like algorithm. To do so, we rely on conditions from Lemma 5.8. To fix notation, for every $\ell < L_3$, let $k[\ell] \in \{1, 2\} \times \{0, 1, 2\}^\ell$ encode $2n \bmod 2 \cdot 3^\ell$.

Let $\rho_{d_\rightarrow} = \prod_{q \in Q^{(\mathrm{o})} \times \{d_\rightarrow\} \times G} \rho_q$ move heads with duck $d_\rightarrow$ right. Still denoting by $d \in \mathrm{Sym}(H)$ the ducking involution and C a condition, the automorphism $f_{d,C} \circ \rho_{d_\rightarrow}^{-1} \circ f_{d,C} \circ \rho_{d_\rightarrow}$ moves head in conveyor belts that verify condition C, and heads with duck $d_\rightarrow$ move to the right (while heads with duck $d_\leftarrow$ move left). Denote this automorphism $\rho_{d_\rightarrow, C}$.

We now proceed as follows. First, apply $(\rho_{d_\rightarrow, \mathrm{len} < L_3})^{-1}$ to move heads with duck $d_\rightarrow$ left, that is, as to go to the last digit of the counter to which we want to add $2n$. Denote by $r' = (0, 1, 2) \in \mathrm{Sym}(\Gamma)$ the addition of a single digit, and $r_{d_\rightarrow} = \mathrm{id} \times \{d_\rightarrow\} \times \mathrm{id} \times r' \in \mathrm{Sym}(H)$. Then, for $0 \leq \ell < L_3$, we do the following.

(a) Add the second digit of $k[\ell]$ on the tape in conveyor belts of length $\ell \leq \mathrm{len} < L_3$: apply $f_{r_{d_\rightarrow}, \ell \leq \mathrm{len} < L_3}$ if $k[\ell]_1 = 1$, or $f_{r_{d_\rightarrow}, \ell \leq \mathrm{len} < L_3}$ if $k[\ell]_1 = 2$.

(b) Perform the carry in conveyor belts of length $\ell \leq \mathrm{len} < L_3$. Let $k' = k[\ell]_{[\![2, |k|-1]\!]}$ be the digits of k we want to check an overflow for, and apply $f_{r_{d_\rightarrow}, (\ell \leq \mathrm{len} < L_3) \wedge (< \bullet k')}$: this applies the carry if the addition on the cells on the right overflowed.

(c) Add the first digit of $k[\ell]$ (that is, $k[\ell]_0 \in \{1, 2\}$) in conveyor belts of length $\mathrm{len} = \ell - 1$. Recall that $Q^{(\mathrm{o})} = \{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\} \times \{1, 2\}$: denoting $b' = (q, 1) \leftrightarrow (q, 2) \in \mathrm{Sym}(Q^{(\mathrm{o})})$ and $b_{d_\rightarrow} = b' \times \{d_\rightarrow\} \times \mathrm{id} \times \mathrm{id} \in \mathrm{Sym}(H)$, apply $f_{b_{d_\rightarrow}, \mathrm{len} = \ell - 1}$ if $k[\ell]_0 = 2$, and the identity otherwise.

(d) Perform the carry in conveyor belts of length $\mathrm{len} = \ell - 1$: apply $f_{b_{d_\rightarrow}, (\mathrm{len} = \ell - 1) \wedge (< k[\ell]_{[\![1, |k[\ell]|-1]\!]})}$, which applies the carry in the state if the addition on the tape overflowed.

(e) Apply $(\rho_{d_\rightarrow, \ell \leq \mathrm{len} < L_3})^{-1}$ to move heads with duck $d_\rightarrow$ left (and heads with duck $d_\leftarrow$ right).

These steps apply the correct addition on heads having duck $d = d_\rightarrow$. On duck $d = d_\leftarrow$, the head simply moves to the right at each step, not modifying anything, until it gets back to its initial positions. All these steps together have word norm $O(L^3)$.

We are left with shifting the tape a times (left or right, depending on the state $\{\blacktriangleright, \blacktriangleleft, \triangleright, \triangleleft\}$), where $a = \lfloor 2n/(2 \cdot 3^\ell) \rfloor$; and apply a final shift if the addition in the previous paragraph overflowed. We do so for each length $0 \leq \ell < L_3$ independently.

Let $s \in \mathrm{Sym}(H)$ be the involution that exchanges states $\blacktriangleright \leftrightarrow \blacktriangleleft$ and $\triangleright \leftrightarrow \triangleleft$. Recall that $\rho_{d_\rightarrow}$ moves every head with duck $d_\rightarrow$ to the right, and that $p_{d_\rightarrow}$ exchanges two adjacent letters if the head has duck $d_\rightarrow$: these let us define $\sigma_{d_\rightarrow} = \prod_{i=1}^{\ell} \rho_{d_\rightarrow} \circ p_{d_\rightarrow}$, the

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

left shift of a whole cyclic tape of length ℓ for states in $\{\blacktriangleright, \triangleright\} \times \{1, 2\} \times \{d_\rightarrow\} \times G$. The automorphism $\sigma_{d_\rightarrow}$ has word norm $O(\ell)$.

Denoting by C_1 the condition len $= \ell$ and C_2 the overflowing condition (that is, the lexicographic comparison $< k[\ell]$), we successively apply $[f_{s,C_1}, (\sigma_{d_\rightarrow})^{-a}]$ (of word norm $O(\ell^2)$) and $[f_{s,C_1 \wedge C_2}, (\sigma_{d_\rightarrow})^{-1}]$ (of word norm $O(\ell^2)$).

Let f_{+2n} be the composition of all these steps. Then f_{+2n} performs the addition of $2n$ in all the conveyor belts of length $< L_3$ with ducks $d_\rightarrow$ simultaneously, is the identity on ducks $d_\leftarrow$, and has word norm $O(L^3)$.

Then, conjugating f_{+2n} by f_{encode} performs $(f_{\mathcal{S}})^{2n}$ on heads with duck $d_\rightarrow$ on conveyor belts of length $< L_3$, and is the identity otherwise. Adding the same automorphism conjugated with f_d and composing (for $d \in \mathrm{Sym}(H)$ the ducking involution), we obtain $(f_{\mathcal{M},\text{len}<L_3})^{2n}$ with word norm $O(L^4)$.

A similar formula exists for $(f_{\mathcal{M},\text{len}\leq L_3})^{2n}$. □

Remark 5.15. The word norm of this implementation of $(f_{\mathcal{S}})^n$ is $O(\log n \cdot (\omega_{\leq}(\log n) + \omega_+(\log n)))$, where $\omega_{\leq}(N)$ is the complexity of the lexicographic inequalities on words of length N, and $\omega_+(N)$ is the complexity of the ternary addition on words of length N. While we could not find a way to perform $\omega_+(N)$ with complexity less than $O(N^3)$, it would be interesting to optimize this specific operation by itself.

6. *Corollaries*

We prove the other theorems listed in the introduction, all of which are straightforward corollaries of Theorem A (and its proof).

6.1. *Distortion in other subshifts*

THEOREM B. *Let X be a sofic shift. Then* $\mathrm{Aut}(X)$ *contains a distortion element if and only if X is uncountable.*

Proof. If X is uncountable, then $\mathrm{Aut}(A^{\mathbb{Z}}) \leq \mathrm{Aut}(X)$ [33, 46]. If X is countable, then the proof of [49, Proposition 2] shows that every automorphism $f \in \mathrm{Aut}(X)$ is either periodic or admits a *spaceship*, namely a configuration of the form $x = ...uuuuvwww...$ which is not spatially periodic, and $f^n(x) = \sigma^m(x)$ for some $m \neq 0$. Clearly this prevents distortion. □

Recall that the lower entropy dimension [37] is

$$\underline{D}(X) = \liminf_{k\to\infty} \frac{\log(\log N_k(X))}{\log k}.$$

We recall and prove Lemma 1.1 (which was used to prove Theorem C).

LEMMA 6.1. *Let X be a subshift with lower entropy dimension less than $1/d$. If $f \in \mathrm{Aut}(X)$ satisfies $|f^n| = O(\log^d n)$, then f is periodic.*

Proof. Suppose we have $|f^n| = O(\log^d n)$ for large n. Then the radius of f^n is also $O(\log^d n)$. It follows that the trace subshift of f has complexity function at most $n \to N_{\lfloor C \log^d n \rfloor}(X)$ for some constant C. If f is not of finite order, by the Morse–Hedlund

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

theorem, we must have $N_{\lfloor C \log^d n \rfloor}(X) > n$ for all n. Substituting $\lfloor e^{\sqrt[d]{n/C}} \rfloor$ for n, we get $N_n(X) \geq e^{\sqrt[d]{an}}$ for some constant $a > 1$. Substituting this lower bound into the definition of lower entropy dimension, we get $\underline{D}(X) \geq (1/d)$. □

6.2. *Distortion in the group of Turing machines.* We recall the definition of the group of Turing machines from [3].

Definition 6.1. Let $n \geq 2$ and $k \geq 1$. Let Y_n be the full shift on n letters, and $X_k = \{x \in \{0, 1, ..., k\}^{\mathbb{Z}} \mid 0 \notin \{x_i, x_j\} \implies i = j\}$. Then

$$\mathrm{RTM}(n, k) = \{f \in \mathrm{Aut}(Y_n \times X_k) \mid f|_{Y_n \times \{0^{\mathbb{Z}}\}} = \mathrm{id}|_{Y_n \times \{0^{\mathbb{Z}}\}}\}.$$

THEOREM D. *Let $n \geq 2$, $k \geq 1$. Then the group of Turing machines* $\mathrm{RTM}(n, k)$ *contains a distortion element; indeed there is a finitely generated subgroup $G = \langle F \rangle$ and an element f such that $|f^n|_F = O(\log^4 n)$.*

Proof. We show that it immediately follows from the main theorem that RTM(18, 96) has a distortion element. We then explain how to conclude this for all $\mathrm{RTM}(n, k)$.

Recall that our automorphisms use the alphabet

$$\Sigma = (\Gamma^2 \times \{+1, -1\}) \sqcup ((Q \times \Gamma) \times \Gamma) \sqcup (\Gamma \times (Q \times \Gamma)),$$

where $\Gamma = \Gamma^{(o)}$ and $Q = Q^{(o)} \times \{d_{\rightarrow}, d_{\leftarrow}\} \times \{+1, -1\} \times [\![0, 2]\!]$.

We may instead view this as

$$(\Gamma^2 \times \{+1, -1\}) \sqcup (Q \times \{\uparrow, \downarrow\} \times \Gamma^2),$$

by grouping $((Q \times \Gamma) \times \Gamma)$ and $(\Gamma \times (Q \times \Gamma))$ together and replacing the choice with an arrow from $\{\uparrow, \downarrow\}$. Next, moving $\{\uparrow, \downarrow\}$ to the state and dropping $\{+1, -1\}$ out of it, we may view this as

$$(\Gamma^2 \times \{+1, -1\}) \sqcup (Q' \times \{+1, -1\} \times \Gamma^2),$$

for a certain set of states Q' with $|Q'| = |Q| = 96$.

Consider the sofic subshift Z where a symbol of $(Q' \times \{+1, -1\} \times \Gamma^2)$ can appear at most once. We clearly have a conjugacy $Z \cong X_{96} \times Y_{18}$, since $|\Gamma^2 \times \{+1, -1\}| = 18$.

It is easy to see that all of the generators F defined in Lemma 5.14 fix Z. Furthermore, our generators only act near the head, so by definition, this restricted action makes them elements of RTM(18, 96). The element $f_{\mathcal{S}}$ coming from the SMART machine clearly has infinite order, since it acts as the SMART machine on infinite configurations. The word norm of $f_{\mathcal{S}}$ with respect to F of course cannot grow faster after restricting these elements to an invariant subspace, so we obtain that the subgroup of RTM(18, 96) generated by F still has a distortion element, and the distortion is at least as bad as on the full shift.

Now, we describe some minor modifications to the main construction that allow to conclude the result for $\mathrm{RTM}(n, k)$. In the construction of the main theorem, in place of the alphabet recalled above, take any finite set S and use instead

$$((\Gamma^2 \times \{+1, -1\}) \sqcup S) \sqcup (Q' \times ((\Gamma^2 \times \{+1, 1\}) \sqcup S)).$$

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Imagining elements of S as new empty conveyor belts of size 1, it is clear how most generators of F should act, as their action is defined by how they act on finite conveyor belts. The element θ does not respect the conveyor belts, but it is also clear how it should act (now that we have moved $\{+1, -1\}$ out of the state onto the tape)—it simply moves all heads.

Now recall that the only use of θ was to make sure that the automorphisms $f_{\tau_{\mathrm{cb}},\to t}$ (respectively $f_{\tau_{\mathrm{cb}},t\leftarrow}$) are in our group. These automorphisms apply the involution $(+1) \leftrightarrow (-1)$ on the sign carried either by $\Gamma^2 \times \{+1, -1\}$ or by a head, at distance t to the right of the heads (respectively left of the heads, respectively both left and right of the heads). The correct extension of these is simply that the flip $(+1) \leftrightarrow (-1)$ does nothing on symbols in S. Then θ allows the implementation of natural analogs of the automorphisms $f_{\tau_{\mathrm{cb}},\to t}$ and $f_{\tau_{\mathrm{cb}},t\leftarrow}$ (with the exact same description).

Next, we recall that the automorphisms $f_{\tau_{\mathrm{cb}},\to t}$ are only used 'through conjugation', that is, they are used during the two-scale trick in very specific situations where we already know the head is on a large conveyor belt, in particular, there are no S-symbols in the affected part. Thus the proof goes through without any modifications.

The introduction of S with $|S| = t$ changes the group of Turing machines from $\mathrm{RTM}(18, 96)$ to $\mathrm{RTM}(18 + t, 96)$, and $\mathrm{RTM}(18 + t, 96)$ clearly embeds in $\mathrm{RTM}(18 + t, 96 + \ell)$ for any $\ell \geq 0$ (by behaving as the identity when the head is in one of the ℓ many new states). In particular, for large enough m, we can pick t, ℓ such that $18 + t = n^m$ and $96 + \ell = kn^m$, to get a distortion element in a subgroup of $\mathrm{RTM}(n^m, kn^m)$. Finally, there is an embedding of $\mathrm{RTM}(n^m, kn^m)$ into $\mathrm{RTM}(n^m, k)$, by considering m-blocks of cells as individual cells, and considering the word on the tape at the origin as part of the state; and then $\mathrm{RTM}(n^m, k)$ embeds into $\mathrm{RTM}(n, k)$ by moving by m steps at once. □

6.3. *Distortion in the Brin–Thompson group mV.* It was shown by Belk and Bleak that classical reversible Turing machines embed in the Brin–Thompson group $2V$. More generally, the group of Turing machines embeds in $2V$, and indeed this embedding is entirely transparent. For this, we recall the *moving-tape model* of Turing machines.

Definition 6.2. Write $\mathrm{RTM}_{\mathrm{fix}}(n, k)$ for the family of homeomorphisms $f : [\![k]\!] \times [\![n]\!]^{\mathbb{Z}} \to [\![k]\!] \times [\![n]\!]^{\mathbb{Z}}$ such that for some radius $r \geq 1$ and local rule $f_{\mathrm{loc}} : \{0, 1\}^r \times \{0, 1\}^r \times [\![k]\!] \to \{0, 1\}^* \times \{0, 1\}^* \times [\![k]\!]$, we have

$$f(xu.vy, a) = (xu'.v'y, b) \text{ whenever } f_{\mathrm{loc}}(u, v, a) = (u', v', b)$$

and for all u, v, $f_{\mathrm{loc}}(u, v) = (u', v', n)$ satisfies $|u'| + |v'| = 2r$.

A proof of the following easy fact was outlined in [3]; one simply translates tape shifts into head movement in the opposite direction.

LEMMA 6.2. *The family of homeomorphisms $\mathrm{RTM}_{\mathrm{fix}}(n, k)$ forms a group under composition, and there is a canonical group isomorphism $\mathrm{RTM}_{\mathrm{fix}}(n, k) \cong \mathrm{RTM}(n, k)$.*

LEMMA 6.3. *The group $\mathrm{RTM}(n, k)$ embeds in the Brin–Thompson group mV for all $m \geq 2, n \geq 2, k \geq 1$.*

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

Proof. The group $2V$ embeds in mV, so it is enough to show this for $m = 2$. This is very similar to the proof by Belk and Bleak [5], and was also essentially stated in [3], so we only outline the proof. First, it is enough to embed $\mathrm{RTM}(n, 1)$, since $\mathrm{RTM}(n, k)$ embeds in $\mathrm{RTM}(n, k + \ell)$ for all $\ell \geq 0$, thus in particular in $\mathrm{RTM}(n, n^m)$ for sufficiently large m, and this group is isomorphic to $\mathrm{RTM}(n, 1)$ (see the end of the proof of Theorem D).

Now pick a complete suffix code $C \subset \{0, 1\}^*$ of cardinality n, and a complete prefix code $D \in \{0, 1\}^*$ of cardinality n. One can uniquely parse any $x.y \in \{0, 1\}^{\mathbb{Z}}$ as $\cdots u_{-2}u_{-1}.v_0v_1v_2 \cdots$ with $u_{-i} \in C$, $v_i \in D$ for all applicable i, which gives a homeomorphism $\phi : \{0, 1\}^{\mathbb{Z}} \to [\![n]\!]^{\mathbb{Z}}$. For $g \in \mathrm{RTM}(n, 1)$, the map g^{ϕ} is easily seen to be in $2V$, so this gives a group-theoretic embedding of $\mathrm{RTM}(n, 1)$ into $2V$. □

Dynamically, the proof gives a topological conjugacy between the natural action of $\mathrm{RTM}(n, 1)$ and the natural action of the subgroup of $2V$ that respects the encoding.

THEOREM E. *The Brin–Thompson group mV contains a distortion element; indeed there is an element f such that $|f^n| = O(\log^4 n)$.*

Proof. The embedding of the group $\mathrm{RTM}(n, k)$ in particular embeds the group where we constructed a distortion element. Adding the finite generating set of mV clearly cannot make the element less distorted. □

7. *Open questions*

Question 7.1. Are there ever distortion elements in $\mathrm{Aut}(X)$ for X a minimal subshift? What about X of zero entropy?

Minimal subshifts are interesting, because at present, we do not know any restrictions on their automorphism groups, yet all known examples are locally virtually abelian. Zero entropy is interesting because on the one hand, there are many known examples of interesting behaviors in their automorphism groups, but [17] shows that exponential distortion is impossible.

Next, it seems worth recalling the remaining parts of [17, Questions 5.1–5.3].

Question 7.2. Are there more natural (in terms of the group structure) subgroups having distortion elements in $\mathrm{Aut}(A^{\mathbb{Z}})$, or even in $\mathrm{Aut}(X)$, where $X \subset A^G$ is an arbitrary subshift on an abelian group G? For example, can we embed the Heisenberg group (or more generally $\mathrm{SL}(3, \mathbb{Z})$), or the Baumslag–Solitar group $\mathrm{BS}(1, n)$?

The Heisenberg group was originally asked about in [33] (though not explicitly due to distortion concerns). One important note about this group is that every (infinite f.g. torsion-free non-abelian) nilpotent group contains a copy of it. Nilpotent groups are considered some of the simplest (in the non-technical sense) kinds of infinite groups after abelian groups; in the case of automorphism groups of subshifts, we can implement a wide variety of behaviors, yet embeddability of nilpotent groups remains a mystery.

Embedding the Baumslag–Solitar group is the same as finding an element of infinite order that is conjugate to a higher power of itself. We believe the SMART machine does not have this property (before or after an embedding into $\mathrm{Aut}(A^{\mathbb{Z}})$), but we have not proved this.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

A slightly more abstract question of interest is whether there exists an amenable subgroup of the automorphism group of a subshift which has distortion elements. One thing amenability rules out is groups that are too large, for example, f.g.-universal subgroups. The Heisenberg group and $\mathrm{BS}(1, n)$ are of course amenable (even solvable).

Question 7.3. Can a one-sided subshift have distortion elements in its automorphism group? Does $\mathrm{Aut}(A^{\mathbb{N}})$ have distortion elements?

Note that $\mathrm{Aut}(A^{\mathbb{N}})$ is simply the subgroup of $\mathrm{Aut}(A^{\mathbb{Z}})$ consisting of automorphisms f such that both f and f^{-1} have 'one-sided radius', that is, $f(x)_i$, $f^{-1}(x)_i$ depend only on $x_{[\![i,i+r]\!]}$ for some r. We do not even know whether one-sided automorphisms of subshifts can have sublinear radius growth.

As mentioned in the introduction, the true word norm growth of our automorphism is between $\Omega(\log n)$ and $O(\log^4 n)$. It would be of great interest to pinpoint the growth up to a multiplicative constant for our automorphism, or for any other automorphism with sublinear growth.

Question 7.4. What are the distortion functions (word norm growth rates) of elements of $\mathrm{Aut}(A^{\mathbb{Z}})$ (or $\mathrm{Aut}(X)$ for more general subshifts)?

Of course, in the case of a non-finitely generated group like $\mathrm{Aut}(A^{\mathbb{Z}})$, the distortion function depends on the finite generating set chosen. While it is of interest to implement distortion functions with respect to subgroups, a more natural object to consider is the directed set of distortion functions with increasing generating sets, and especially the eventual behavior as the generating set increases.

Similar questions can be asked about groups of Turing machines and the Brin–Thompson $2V$, where we also exhibit elements whose word norm grows polylogarithmically, but have no further control on the distortion.

A natural idea for getting control over the distortion function would be to use, in place of SMART, a general-purpose Turing machine, which is made to have sublinear movement using the reversible Hooper trick from [32] (and finally embedded in some natural way into the automorphism group of a subshift). It is known that this construction always produces Turing machines with zero Lyapunov exponents, that is, with sublinear movement [28, 31].

Question 7.5. Does the Kari–Ollinger construction in [32] always produce distortion elements?

Acknowledgements. We would like to thank Anthony Genevois for pointing out Corollary 1.2. We thank Pierre Guillon for helpful discussions.

References

[1] E. Alibegović. Translation lengths in out (f_n). *Geom. Dedicata* **92**(1) (2002), 87–93.
[2] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, Cambridge, 2009.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press

[3] S. Barbieri, J. Kari and V. Salo. The group of reversible Turing machines. *Cellular Automata and Discrete Complex Systems (Lecture Notes in Computer Science, 9664)*. Eds. M. Cook and T. Neary. Springer, Cham, 2016, pp. 49–62.
[4] D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *J. Comput. System Sci.* **38**(1) (1989), 150–164.
[5] J. Belk and C. Bleak. Some undecidability results for asynchronous transducers and the Brin–Thompson group $2V$. *Trans. Amer. Math. Soc.* **369**(5) (2017), 3157–3172.
[6] N. Bitar, S. Donoso and A. Maass. Distortion in automorphisms of expansive systems. *Automata and Complexity: Essays Presented to Eric Goles on the Occasion of His 70th Birthday*. Ed. A. Adamatzky. Springer International Publishing, Cham, 2022, pp. 21–41.
[7] V. D. Blondel, J. Cassaigne and C. Nichitiu. On the presence of periodic configurations in turing machines and in counter machines. *Theoret. Comput. Sci.* **289**(1) (2002), 573–590.
[8] T. Boykett. Closed systems of invertible maps. *J. Mult.-Valued Logic Soft Comput.* **32**(5–6) (2019), 565–605.
[9] T. Boykett, J. Kari and V. Salo. Finite generating sets for reversible gate sets under general conservation laws. *Theoret. Comput. Sci.* **701** (2017), 27–39.
[10] M. Boyle, D. Lind and D. Rudolph. The automorphism group of a shift of finite type. *Trans. Amer. Math. Soc.* **306**(1) (1988), 71–114.
[11] M. G. Brin. Higher dimensional Thompson groups. *Geom. Dedicata* **108** (2004), 163–192.
[12] D. Calegari and M. H. Freedman. Distortion in transformation groups. *Geom. Topol.* **10** (2006), 267–293, with an appendix by Yves de Cornulier.
[13] S. Cantat and Y. de Cornulier. Distortion in Cremona groups. *Ann. Sc. Norm. Super. Pisa Cl. Sci. (5)* **20**(2) (2020), 827–858.
[14] J. Cassaigne, N. Ollinger and R. Torres-Avilés. A small minimal aperiodic reversible Turing machine. *J. Comput. System Sci.* **84** (2017), 288–301.
[15] P. Concha-Vega and R. Torres-Avilés. A binary complete and aperiodic Turing machine. *Int. J. Unconv. Comput.* **16**(1) (2021), 19–39.
[16] V. Cyr, J. Franks and B. Kra. The spacetime of a shift endomorphism. *Trans. Amer. Math. Soc.* **371**(1) (2019), 461–488.
[17] V. Cyr, J. Franks, B. Kra and S. Petite. Distortion and the automorphism group of a shift. *J. Mod. Dyn.* **13**(1) (2018), 147–161.
[18] V. Cyr and B. Kra. The automorphism group of a minimal shift of stretched exponential growth. *J. Mod. Dyn.* **10**(02) (2016), 483.
[19] T. C. Davis and A. Y. Olshanskii. Subgroup distortion in wreath products of cyclic groups. *J. Pure Appl. Algebra* **215**(12) (2011), 2987–3004.
[20] S. Donoso, F. Durand, A. Maass and S. Petite. On automorphism groups of low complexity subshifts. *Ergod. Th. & Dynam. Sys.* **36**(1) (2016), 64–95.
[21] B. Farb, A. Lubotzky and Y. Minsky. Rank-1 phenomena for mapping class groups. *Duke Math. J.* **106**(3) (2001), 581–597.
[22] D. S. Farley. Actions of picture groups on CAT(0) cubical complexes. *Geom. Dedicata* **110** (2005), 221–242.
[23] J. Franks and M. Handel. Distortion elements in group actions on surfaces. *Duke Math. J.* **131**(3) (2006), 441–468.
[24] Ś. R. Gal and J. Kedra. On distortion in groups of homeomorphisms. *J. Mod. Dyn.* **5**(3) (2011), 609–622.
[25] S. M. Gersten and H. B. Short. Rational subgroups of biautomatic groups. *Ann. of Math. (2)* **134** (1991), 125–158.
[26] M. Gromov. Asymptotic invariants of infinite groups. *Geometric Group Theory. Volume 2 (Sussex, 1991) (London Mathematical Society Lecture Note Series, 182)*. Ed. G. A. Niblo. Cambridge University Press, Cambridge, 1993, pp. 1–295.
[27] N. Guelman and I. Liousse. Distortion in groups of affine interval exchange transformations. *Groups Geom. Dyn.* **13**(3) (2019), 795–819.
[28] P. Guillon and V. Salo. Distortion in one-head machines and cellular automata. *Cellular Automata and Discrete Complex Systems – 23rd IFIP WG 1.5 International Workshop, AUTOMATA 2017, Milan, Italy, June 7–9, 2017, Proceedings (Lecture Notes in Computer Science, 10248)*. Eds. A. Dennunzio, E. Formenti, L. Manzoni and A. E. Porreca. Springer, Cham, 2017, pp. 120–138.
[29] F. Haglund. Isometries of CAT(0) cube complexes are semi-simple. *Ann. Math. Qué.* (2021), doi:10.1007/s40316-021-00186-2.
[30] P. K. Hooper. The undecidability of the Turing machine immortality problem. *J. Symb. Log.* **31**(2) (1966), 219–234.

[31] E. Jeandel. Computability of the entropy of one-tape Turing machines. *Proc. 31st Int. Symp. on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5–8, 2014, Lyon, France (LIPIcs, 25)*. Eds. E. W. Mayr and N. Portier. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, 2014, pp. 421–432.
[32] J. Kari and N. Ollinger. Periodicity and immortality in reversible computing. *Mathematical Foundations of Computer Science 2008*. Eds. E. Ochmański and J. Tyszkiewicz. Springer, Berlin–Heidelberg, 2008, pp. 419–430.
[33] K. H. Kim and F. W. Roush. On the automorphism groups of subshifts. *Pure Math. Appl. (PU.M.A.)* **1**(4) (1990), 203–230.
[34] J. Kopra and V. Salo. Sofically presented dynamical systems. *Preprint*, 2021, arXiv:2105.06767.
[35] P. Kůrka. On topological dynamics of Turing machines. *Theoret. Comput. Sci.* **174**(1–2) (1997), 203–216.
[36] F. Le Roux and K. Mann. Strong distortion in transformation groups. *Bull. Lond. Math. Soc.* **50**(1) (2018), 46–62.
[37] T. Meyerovitch. Growth-type invariants for $\mathbb{Z}^d$ subshifts of finite type and arithmetical classes of real numbers. *Invent. Math.* **184**(3) (2011), 567–589.
[38] K. A. Mihaĭlova. The occurrence problem for free products of groups. *Mat. Sb. (N.S.)* **75**(117) (1968), 199–210.
[39] C. Moore. Generalized shifts: unpredictability and undecidability in dynamical systems. *Nonlinearity* **4**(2) (1991), 199.
[40] A. Navas. (Un)distorted diffeomorphisms in different regularities. *Israel J. Math.* **244**(2) (2021), 727–741.
[41] N. Ollinger. On aperiodic reversible Turing machines (invited talk). *Reversible Computation: 10th International Conference (RC 2018) (Lecture Notes in Computer Science, 11106)*. Eds. J. Kari and I. Ulidowski. Springer, Cham, 2018, pp. 61–64.
[42] O. Ore. Some remarks on commutators. *Proc. Amer. Math. Soc.* **2**(2) (1951), 307–314.
[43] R. Pavlov and S. Schmieding. Local finiteness and automorphism groups of low complexity subshifts. *Ergod. Th. & Dynam. Sys.* **43** (2023), 1980–2001.
[44] M. Pengitore. Residual finiteness and strict distortion of cyclic subgroups of solvable groups. *J. Algebra* **546** (2020), 679–688.
[45] Y. V. Rogozhin. Undecidability of the immortality problem for Turing machines with three states. *Cybernetics*, **11**(1) (1975), 46–49.
[46] V. Salo. A note on subgroups of automorphism groups of full shifts. *Ergod. Th. & Dynam. Sys.* **38**(4) (2018), 1588–1600.
[47] V. Salo. Graph and wreath products of cellular automata. *Preprint*, 2020, arXiv:2012.10186.
[48] V. Salo. Universal groups of cellular automata. *Colloq. Math.* **169**(1) (2022), 39–77.
[49] V. Salo and I. Törmä. Computational aspects of cellular automata on countable sofic shifts. *Mathematical Foundations of Computer Science 2012 (Lecture Notes in Computer Science, 7464)*. Eds. B. Rovan, V. Sassone and P. Widmayer. Springer, Heidelberg, 2012, pp. 777–788.
[50] S. Schmieding. Automorphisms of the shift: Lyapunov exponents, entropy, and the dimension representation. *Ergod. Th. & Dynam. Sys.* **40**(9) (2020), 2552–2570.
[51] P. Selinger. Reversible k-valued logic circuits are finitely generated for odd k. *Preprint*, 2016, arXiv:1604.01646.

https://doi.org/10.1017/etds.2023.67 Published online by Cambridge University Press