Numerical upper bounds on growth of automata groups

Jérémie Brieussel, Thibault Godin, Bijan Mohammadi

Numerical upper bounds on growth of automata groups

Jérémie Brieussel[1], Thibault Godin[1,2,3], and Bijan Mohammadi[1]

[1]Université de Montpellier – Institut Montpelliérain Alexander Grothendieck
[2]Université de Lorraine – Institut Elie Cartan de Lorraine
[3]University of Turku – Department of Mathematics

Abstract

The growth of a finitely generated group is an important geometric invariant which has been studied for decades. It can be either polynomial, for a well-understood class of groups, or exponential, for most groups studied by geometers, or intermediate, that is between polynomial and exponential. Despite recent spectacular progresses, the class of groups with intermediate growth remains largely mysterious. Many examples of such groups are constructed using Mealy automata. The aim of this paper is to give an algorithmic procedure to study the growth of such automata groups, and more precisely to provide numerical upper bounds on their exponents.

Our functions retrieve known optimal bounds on the famous first Grigorchuk group. They also improve known upper bounds on other automata groups and permitted us to discover several new examples of automata groups of intermediate growth. All the algorithms described are implemented in `GAP`, a language dedicated to computational group theory.

1 Introduction

The aim of this paper is to develop tools to analyse the growth of automata groups, and more precisely to provide numerical upper bounds when the growth is intermediate.

Groups are fundamental objects in mathematics, which can be seen as an abstract encoding of the notion of symmetry. They are used in various domains of discrete mathematics and combinatorics, e.g. graph isomorphism [33]), cryptography [12], physics [51], chemistry [18] and even biology [20]. The ideal goal of group theory is to describe and classify all the possible behaviours that a group can exhibit. A landmark result in this direction is the *classification of finite simple groups* [17]. For infinite groups, various interesting phenomena can appear. For instance a group can be infinite and yet have all its elements of finite order. Such groups are called Burnside groups in homage to Burnside who asked whether they exist in 1902. First examples were given in 1964 by Golod and Shafarevich [25, 26]. Subsequent examples were given in 1968 by Novikov and Adyan [46] and in 1972 by Aleshin [2], the later using automata groups.

An geometric way to classify groups is through their growth: let G be a finitely generated group and S a generating set of G. The *Cayley graph* of (G, S) is the graph $\Gamma_{G,S}$ whose vertices are the elements of G and such that there is an oriented edge from g to h if there exists $s \in S$ such that $g.s = h$. Such an edge is naturally labelled by s. Examples of such graphs are depicted in Fig. 1. The *growth* function of a group G with respect to S is the function $\gamma_{G,S}(\ell)$ that counts the number of elements in a ball of given radius ℓ in $\Gamma_{G,S}$. This notion was introduced by Svarc [52] and Milnor [38] in relation with Riemannian geometry. They observed that the growth of a group is essentially independent of the generating set (see Lemma 3.3). Obviously the growth is bounded

if and only if the group is finite. Moreover abelian (i.e. commutative) groups have polynomial growth. For instance, integer lattices $\mathbb{Z}^d$ have polynomial growth of degree d. This is still the case for the (slightly bigger) class of nilpotent groups [57], and a celebrated result of Gromov asserts that there are essentially no other such groups, since polynomial growth implies nilpotency up to taking a finite index subgroup [32].
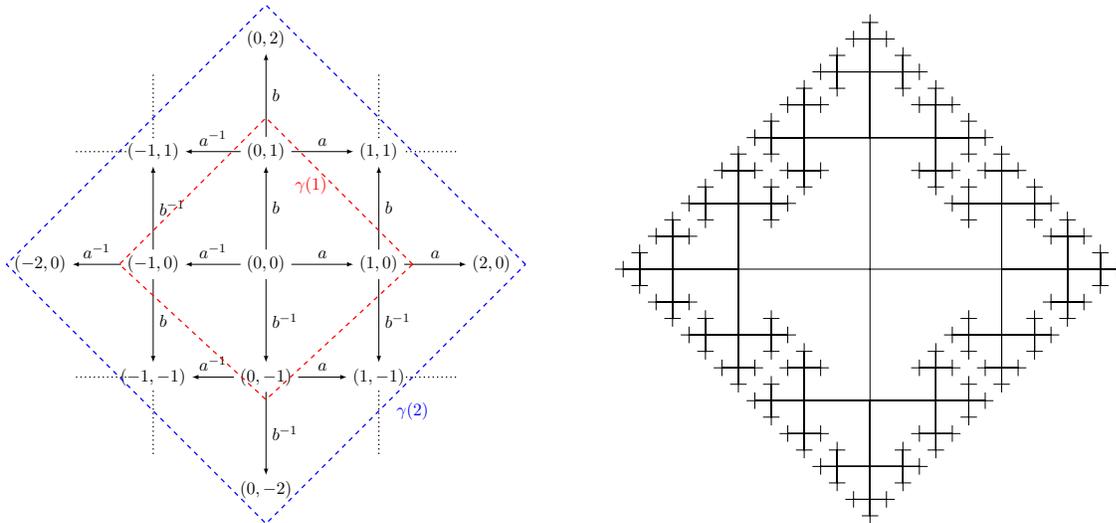

Figure 1: On the left: the Cayley graph of $\mathbb{Z}^2$ with generators $a^{\pm 1} = (0, \pm 1)$ and $b^{\pm 1} = (\pm 1, 0)$ together with the balls of radii 1 et 2. On the right: the ball of radius 5 in the (unlabelled) Cayley graph of the free group on 2 generators.

On the other hand, free groups, whose Cayley graphs are essentially infinite regular trees, and non-elementary hyperbolic groups have exponential growth. Among several classes of groups, the growth is either polynomial or exponential. This dichotomy holds for instance among solvable groups by Milnor and Wolf [37, 57], or among linear groups, i.e. groups of invertible matrices, by the Tits alternative [53]. We refer the reader to more complete introduction to this topic in [19, 31, 36, 30].

Motivated by such observations, Milnor asked in 1968 if this picture was complete or if there exists groups having growth functions between polynomial and exponential, henceforth called *groups of intermediate growth* [39]. This question was positively answered by Grigorchuk in the eighties:

Theorem 1.1 ([27, 28],[3, 22])**.** *The group generated by the Mealy automaton of Fig. 2 has intermediate growth. Its growth function is essentially equivalent to* $\exp(\ell^{0.7674\cdots})$*.*

The computation of the growth function of this group is challenging. Grigorchuk proved that it is between $e^{\sqrt{n}}$ and $e^{\ell^{0.993}}$ in [28]. Bartholdi gave an accurate upper bound [3]. Only recently Erschler and Zheng provided a matching (up to logarithmic factor) lower bound [22].

The group generated by the Mealy automaton depicted in Fig. 2 is now called the *first Grigorchuk group*. Mealy automata are a special class of transducers (automata with input and output) whose functions induced by states on words generate groups. These have been ubiquitous in growth related questions since the eighties: most constructions of groups of intermediate growth use them [28, 29, 23, 11, 4, 14, 8, 16] they are a basic tool to construct groups of given intermediate growth [9, 15] and they were also used by Wilson [56] to exhibit groups having non-uniform exponential growth (answering a question of Gromov). Only recently Nekrashevych gave a construction of groups of intermediate growth not based on the use of Mealy automata [45].

This new construction is very different since Nekrashevych's groups are *simple* (i.e. have no non-trivial normal subgroups) whereas automata groups are residually finite (in particular, they have infinitely many normal subgroups of finite index).

In spite of this list of examples, few things are actually known about the growth of general automata groups: some restricted classes of automata generate only groups with bounded or exponential growth [34, 47]. It contrasts with the usual amenability of automata groups to algorithmic questions [6, 1, 35, 10, 24, 5, 43].

Definition 1.2. A number $\alpha \in [0, 1]$ is called an *upper bound exponent* of the group G if for any generating set S there exists $c > 0$ such that $\gamma_{G,S}(\ell) \leq \exp(c\ell^\alpha)$ for all ℓ.

It is an open question whether there exists groups with growth exponent smaller than 0.7674.... Grigorchuk's gap conjecture asserts that there are no groups with growth below $\exp(\sqrt{\ell})$. So far, only the groups in [28, 29, 11, 16] (and related constructions) are known to have an exponent < 1. It is likely that the groups in [23, 4, 14] have intermediate growth but exponent 1 as their growth rate could be essentially $\exp(\frac{\ell}{\log(\ell)})$. Such a behavior is known to hold for the group $G_{(01)^\infty}$ also introduced by Grigorchuk [28] and studied by Erschler [21]. Nekrashevych conjectures that some groups in [45] have growth exponent below 0.7674..., but it is not yet proved that they have exponent < 1. Regarding automata groups, the only accurate computations of exponents concern Grigorchuk groups acting on the binary tree. They appeared in a recent work of Erschler and Zheng [22].

The present work aims at giving explicit numerical upper bounds on the growth of automata groups and to yield new examples of such groups.

Proposed approach: All known examples of automata groups of intermediate growth belong to the class of *(strongly) contracting* automata. More precisely, the contraction method described in Section 5 and originaly due to Grigorchuk [27, 28, 29] is the only strategy known so far to obtain non-trivial upper bounds for growth exponents.

An element g of an automaton group $\langle \mathcal{A} \rangle$ acting on Σ^* can be recursively described as $g = \left\langle g_{|x_1}, \ldots, g_{|x_d} \right\rangle \pi(g)$ where $\pi(g)$ is a permutation of Σ and the $g_{|x}$ are elements of the group called the *sections* of g. We say that g is *(strongly) contracting* if the length of g is strictly smaller than the sum of the length of its section, and we set $\eta(g)$ to be their ratio.

The contraction method ensures that if we find a suitable finite set Δ containing only contracting elements, the group has subexponential growth bounded above by $\exp(\ell^\alpha)$, where α depends explicitly on the ratio η and the size of the alphabet. Suitable sets have to be *essentially geodesic generating*, in short eggs. This property is defined in Section 4.

In practice, given a target η, we try to find such an egg Δ. For this, we focus on an over-approximation of $\langle \mathcal{A} \rangle$ in the sense that we perform our computations in a group $\tilde{G}$ with the same generators as $\langle \mathcal{A} \rangle$ but only a finite number of relations, so that $\langle \mathcal{A} \rangle$ is a quotient of $\tilde{G}$, and explore the Cayley graph of the latter group. The basic idea is to implement a graph traversal from the identity, stopping the exploration of a branch when we have found an element which is contracting enough or which has already been visited. Using a tight over-approximation allows to accelerate computations but in balance we have to keep a good understanding of the Cayley graph structure in order to guarantee that the computed bound is still relevant in the original automaton group – see Section 4.2. This first (semi-)algorithm already presents an interesting combinatorics as the results may vary substantially according to the target.

In a second time, we introduce weights in computations, as it was successfully done by Bartholdi for the first Grigorchuk group [3]. Weights are positive functions π on the generating set S, and

we declare the length of an edge to equal the weight of the generator that labels it. This gives a new, non-uniform and possibly non-symmetric, metric on the Cayley graph. It is well-known that the growth rate is not modified by the introduction of weights, nor by change of generating set – see Lemma 3.3. The upper bound α obtained by the contraction method depends drastically on the weight π as can be seen on Tables 1 and 3, but by invariance the result applies to all growth functions $\gamma_{G,S,\pi}(\ell)$.

A crucial issue is to find the best possible weights for a given generating set. For this, we fix an essentially geodesic generating set Δ and optimize the returned contraction parameter $\eta(\pi)$ under the condition that Δ remains an egg for π, which is the case under mild triangular conditions described in Section 4.3. In practice, we are lead to minimize a function of the form $\max_{w \in \Delta}(f_w)$, where the functions f_w are rational (homogeneous) functions on $\mathbb{R}^d$, on a subdomain of $[0,1]^d$ bounded by linear inequalities. For this, we use a numerical algorithm wmo based on a generalized gradient method described in [40, 41] – see Section 6.2.

Finally, using the weight-metric-optimization algorithm wmo, we develop a dynamical procedure trying to optimize the contraction through both the graph-search of an egg Δ and the choice of the weights π. Notice that these two possible directions of optimization, by graph exploration and weights choice, strongly interact. Hence, a careful tuning of the parameters might be needed for some groups. On the other hand, pre-set routines often give rather good results– see Tables 2 and 3.

Previous works: As mentionned above, the underlying automaton allows the class of automata groups to be efficiently studied from an algorithmic point of view. For instance the word problem can be efficiently solved as well as the order problem [13] or the Engel problem [6] in some specific classes. Two main packages have been developed to this end: AutomGrp [43] by Muntyan and Savchuk and FR [5] by Bartholdi. Both are implemented in the GAP language [24], specialized in computational algebra.

The situation is more contrasted for growth related question: a first experimental approach was attempt by Reznykov and Sushchanskii [49], where the growth function up to a certain length is computed, which allows the author to deduce conjectures from the observed values. Using this approach, they exhibit the smallest automaton semigroup with intermediate growth. Yet, no exact result can be inferred from this work and, as mentionned in [48], the first terms of the growth function can be misleading. For instance, it follows from [9, 15] that the growth may look exponential on large balls and still be intermediate at even larger scales. Notice that an efficient algorithm to compute the growth of automata groups, based on minimization, is provided by [35].

In FR, no function trying to determine the growth behaviour of an automaton group is provided (the functions computing the polynomial or exponential growth degree are dealing with a different notion, that of activity introduced by Sidki [50]).

In the package AutomGrp, a function testing if the automaton generates a group of subexponential growth is proposed. The tactic used is also based on the contraction method and can be seen as the simplest version of our algorithm: it looks if there exists a length ℓ such as every words of length ℓ are contracting. Our present approach refines this in several ways.

First we no longer look for elements of the same length, but rather use the concept of essentially geodesic generating sets, introduced in Section 4. It permits to reduce drastically the number of elements whose contraction is required in order to assure subexponentiality. This allows our algorithm to detect groups of intermediate growth that are missed by AutomGrp, such as the group generated by the automaton Fig. 8.

Secondly, we use weights in the computation of contraction and actually implement an algorithm to numerically optimize them. Note that our dynamical procedures also provide an explicit bound when the procedure stops (but the bound is possibly trivial). We stress that our code relies on

level	weights chosen	target η	obtained η	obtained α
1	[.25, .25, .25, .25]	.99	does not end	does not end
	[.25, .375, .25, .125]	.99	.875	.8385
	[.305061, .34747, .223839, .123631]*	.99	.8106*	.7675*
2	[.25, .25, .25, .25]	.99	does not end	does not end
	[.25, .375, .25, .125]	.99	.9231	.9455
	[.305061, .34747, .223839, .123631]*	.99	.9152	.9400
	[.305061, .34747, .223839, .123631]*	.75	.7497	.8280
	[.305061, .34747, .223839, .123631]*	.68	.6800	.7824
3	[.25, .25, .25, .25]	.99	.8889	.9437
	[.305061, .34747, .223839, .123631]*	.99	.8954	.9496
	[.305061, .34747, .223839, .123631]*	.75	.7418	.8745
	[.305061, .34747, .223839, .123631]*	.58	.5800	.7924

Table 1: The first Grigorchuk group: upper bounds computed by function `IsSubExp_rec` without optimization. The stars $*$ mark optimal and almost optimal data. The optimal weights are taken from Bartholdi [3] and have been normalized to sum up to 1. Levels are defined at the end of Section 2.1.

level	update	target η	obtained η	obtained weights	obtained α
1	4	.90	.8107*	[.3052, .3475, .2243, .1236]*	.7676*
2	4	.90	.8121	[.3072, .3465, .1229, .2236]	.8063
	4	.72	.7166	[.3069, .3466, .2068, .1399]	.9464
3	4 or 10	.90	.8889	[.25, .25, .25, .25]	.9464
	4	.65	.6477	[.4789, .2606, .2059, .0548]	.8273
	10	.63	.6287	[.4004, .2999, .2345, .0654]	.8176

Table 2: The first Grigorchuk group: upper bounds computed by function `IsSubExp_opt` with optimization of weights every update rounds, starting with uniform weights. The stars $*$ mark optimal and almost optimal data.

Automaton	obtained η	obtained α	obtained weights	comments
Fig. 7	.6450	.8034	[.3352, .1899, .1899, .2849]	New automaton group weights obt. by optimization
Fig. 8, from [16]	.8188	.9123	[1., 0., 0.]	prev. bound from [16]
	.8300	.9178	[.25, .25, .25, .25]	was .9396
Fig. 9	.8398	.9177	[.3785, .2655, .3561]	New automaton group weights obt. by optimization

Table 3: Some new upper bounds computed.

some functions developed in `AutomGrp`, the most important being the rewriting system for groups.

Results obtained: Our approach produces several positive results:

- *On the first Grigorchuk group:* our algorithm allows to retrieve numerically the optimal bound and weight system, see Tables 1 and 2. Optimality of $\alpha = 0.7674...$ is known from [22]. Throughout the paper, the numerical values we obtain are rounded up to the fourth digit.

- *Improvement of upper bounds:* for the automata in [16], depicted in Fig. 8, we improve the best known upper bound from 0.9396 to 0.9123, see Table 3.

- *New groups of intermediate growth:* we discovered several new automata generating group of intermediate growth (see Figs. 7 and 9 and Table 3.). It should be noticed that these are not just twists of known automata but rather seem to belong to new families.

Perspectives: Our approach can be naturally extended to (homogeneous) spinal groups [11] which generalize Grigorchuk groups and are known to be of intermediate growth. Moreover, the new groups of intermediate growth described in Figs. 7 to 9 seem to belong to large families of groups with intermediate growth. A systematic study of these families is in order. One aim of the present paper was to provide tools for this study.

From a computational point of view, we can aim for two main improvements: firstly, the procedures could be implemented in an other, more efficient, language. Secondly, the procedure of exploration is tractable to parallelization, which would greatly improve the performances of our methods.

Finally, we stress that many algorithmic problems are undecidable for automata groups, and that it is unknown if there exists an algorithm proving that an automaton is contracting. On the other hand, some decidable classes of automata, e.g. the subclass of bounded automata [44, Theorem 6.5] which contains most known automata groups of intermediate growth, force the generated group to be contracting in a sense. It would be interesting to explore these classes with our algorithms. Moreover, one can wonder if the growth rate is decidable in the class of bounded automata, which contains groups of all types of growths.

Organization of the paper. Basic facts about Mealy automata and their associated groups are briefly presented in Section 2, together with the examples under focus in the paper. Word metric and growth rates are defined in Section 3. The crucial notion of eggs (essentially geodesic generating sets) is defined and studied in Section 4. The strong contraction method for subexponential growth is described in Section 5. Section 6 is devoted to the description of (semi)-algorithms that provide upper bounds. Experimental data obtained with them and comments appear in Section 7. The last Section 8 gives a criterion ensuring that the groups we consider do not have polynomial growth. Appendix A gathers the pictures of diagrams and Schreier graphs of all automata groups studied here. The `GAP` code is available at `https://www.irif.fr/~godin/automatongrowth.html`.

2 Automata groups

We briefly review definitions and elementary facts about Mealy automata and self-similar group. The reader is refered to [44, 58] for comprehensive description.

2.1 Mealy automata

A *Mealy automaton* is a complete deterministic letter-to-letter transducer $\mathcal{A} = (Q, \Sigma, \delta, \rho)$, where Q and Σ are finite sets respectively called the the *stateset* and the *alphabet*, and $\delta = (\delta_i : Q \to Q)_{i \in \Sigma}$, $\rho = (\rho_q : \Sigma \to \Sigma)_{q \in Q}$ are respectively called the *transition* and *production* functions. Examples of such Mealy automata are depicted Figure 2. The transition and production functions can be extended to words as follows: see $\mathcal{A}$ as an automaton with input and output tapes, thus defining mappings from input words over Σ to output words over Σ. Formally, for $q \in Q$, the map $\rho_q \colon \Sigma^* \to \Sigma^*$, extending $\rho_q \colon \Sigma \to \Sigma$, is defined recursively by:

$$\forall i \in \Sigma, \ \forall \mathbf{s} \in \Sigma^*, \qquad \rho_q(i\mathbf{s}) = \rho_q(i)\rho_{\delta_i(q)}(\mathbf{s}) \,. \tag{1}$$

Observe that ρ_q preserves the length of words in Σ^*. We can also extend the map ρ to words of states $\mathbf{u} \in Q^*$ by composing the production functions associated with the letters of $\mathbf{u}$:

$$\forall q \in Q, \ \forall \mathbf{u} \in Q^*, \qquad \rho_{q\mathbf{u}} = \rho_{\mathbf{u}} \circ \rho_q \,. \tag{2}$$

Therefore the production functions $\rho_q : \Sigma^* \to \Sigma^*$ of an automaton $\mathcal{A}$ generate a semigroup $\langle \mathcal{A} \rangle_+ := \{\rho_{\mathbf{u}} : \Sigma^* \to \Sigma^* | \mathbf{u} \in Q^*\}$, all of which elements preserve the length of words in Σ^*.

A Mealy automaton is said to be *invertible* whenever ρ_q is a permutation of the alphabet for every $q \in Q$. In this case, all functions $\rho_q : \Sigma^* \to \Sigma^*$ are invertible and the automaton actually generates a group:

$$\langle \mathcal{A} \rangle := \langle \rho_q^{\pm 1} : \Sigma^* \to \Sigma^* | q \in Q \rangle = \left\{ \rho_{\mathbf{u}} : \Sigma^* \to \Sigma^* | \mathbf{u} \in \left(Q \cup Q^{-1} \right)^* \right\} \,.$$

It is not difficult to find another (symmetrized) Mealy automaton generating this group as a semigroup. Its stateset can be taken as $Q \cup Q^{-1}$ such that $\rho_q^{-1} = \rho_{q^{-1}}$. A group of the form $\langle \mathcal{A} \rangle$ for an invertible automaton is called an *automaton group*. Such a group acts naturally on the set Σ^*.

Observe that given an integer k and a Mealy automaton $\mathcal{A} = (Q, \Sigma, \delta, \rho)$ with alphabet Σ, one can associate to it a Mealy automaton $(Q, \Sigma^k, \delta, \rho)$ where we identify the function ρ_q acting on Σ with the same function acting on Σ^k via (1). This just amounts to replacing the alphabet by the set of syllables of length k. With a slight abuse of language, we call the latter automaton the level k of $\mathcal{A}$. Notice that this level k automaton generates the same group as the original automaton.

2.2 Self-similar groups

One can describe the automaton group through this action via *wreath recursion* and obtain a so-called *self-similar* group. First let us recall a few definitions. Let us write $\Gamma \curvearrowright X$ the action of a group Γ on a set X and $\gamma.x \in X$ the transformation of $x \in X$ induced by $\gamma \in \Gamma$. The *permutational wreath product* $G \wr_\Sigma \mathrm{Sym}(\Sigma)$ of a group G over a set Σ is the set $G^\Sigma \times \mathrm{Sym}(\Sigma)$ together with the operation

$$\left\langle g_{|x_1}, \ldots, g_{|x_d} \right\rangle \pi(g) . \left\langle h_{|x_1}, \ldots, h_{|x_d} \right\rangle \pi(h) = \left\langle g_{|x_1} h_{|\pi(g)^{-1}.x_1}, \ldots, g_{|x_d} h_{|\pi(g)^{-1}.x_d} \right\rangle \pi(g)\pi(h)$$

for the obvious action of $\mathrm{Sym}(\Sigma)$ on $\Sigma = \{x_1, x_2, \ldots, x_d\}$.

7

Assume we are given a group G together with an injective homomorphism $\psi : G \hookrightarrow G \wr_\Sigma \mathrm{Sym}(\Sigma)$. Then one can describe the group elements with a recursive formula, where we canonically identify g and $\psi(g)$

$$g = \psi(g) = \left\langle g_{|x_1}, \ldots, g_{|x_d} \right\rangle \pi(g) \,. \tag{3}$$

The group element $g_{|x_i}$ is called the *section* of g in x_i and $\pi(g)$ is the *action* of g. One can extend the section to words : $\forall \mathbf{u}x \in \Sigma^*, g_{|\mathbf{u}x} = g_{|\mathbf{u}|x}$.

One can also extend the action of G to Σ^* by $\forall x\mathbf{u} \in \Sigma^*, g.x\mathbf{u} = (\pi(g).x)g_{|x}.\mathbf{u}$. The action of G on Σ^*, is called *self-similar* if

$$\forall g \in G, \forall \mathbf{u} \in \Sigma^*, \forall x \in \Sigma, \exists h \in G, \exists y \in \Sigma, g.x\mathbf{u} = yh.\mathbf{u} \,.$$

If, for every element g of G the set $\{g_{|\mathbf{u}}, \mathbf{u} \in \Sigma^*\}$ is finite, the action is said to be *finite state*. It is easy to see that such a finite state self-similar action corresponds to exactly one automaton group, possibly with infinite stateset. A quick way to describe an automaton group is to give the section decomposition (3) of all elements of the stateset.

The correspondence between Mealy automata and self-similar groups can be summarized as:

$$\rho_q(x) = y \text{ and } \delta_x(q) = p \quad \Longleftrightarrow \quad q_{|x} = p \text{ and } \pi(q).x = y \,,$$

which we represent by the arrow notation $q \xrightarrow{x|y} p \in \mathcal{A}$. The stateset of the automaton corresponds to a generating set of the self-similar group.

2.3 Diagram, dual automaton and Schreier graph

The arrow representation permits to describe a Mealy automata by its *diagram*, which is a labelled oriented graph. The vertices of the diagram are labelled by the states. From each vertex q and letter x, there is an oriented edge from q to $p = \delta_x(q)$ which we label by $x \mid y$ where $y = \rho_q(x)$. The diagrams of most groups studied in this paper are pictured in Appendix A.

The roles of the stateset and the alphabet in an automaton are symmetric in the definition. The *dual automaton* of $\mathcal{A} = (Q, \Sigma, \delta, \rho)$ is the automata $\hat{\mathcal{A}} = (\Sigma, Q, \rho, \delta)$ obtained by exchanging them. This amounts to

$$q \xrightarrow{x|y} p \in \mathcal{A} \quad \Longleftrightarrow \quad x \xrightarrow{q|p} y \in \hat{\mathcal{A}} \quad \text{for all } p, q, x, y.$$

When we draw the diagram of the dual automaton $\hat{\mathcal{A}}$, we obtain the *Schreier graph* of the action of $\langle \mathcal{A} \rangle$ on the alphabet Σ, i.e. the graph with vertex set Σ and edges from x to $y = q.x = \rho_q(x)$ labelled by q.

2.4 Examples

In this paper we will focus on a few iconic examples.

- The first Grigorchuk group is the most famous group of intermediate growth. A stateset is given by
$$a = \langle e, e \rangle (0, 1), \quad b = \langle a, c \rangle, \quad c = \langle a, d \rangle, \quad d = \langle e, b \rangle.$$

  Its diagram and Schreier graphs are depicted in Fig. 2. We also apply our algorithm to disguised versions of the first Grigorchuk, where we look at it as generated by the automata acting on the second or third levels of the tree. They provide interesting situations where
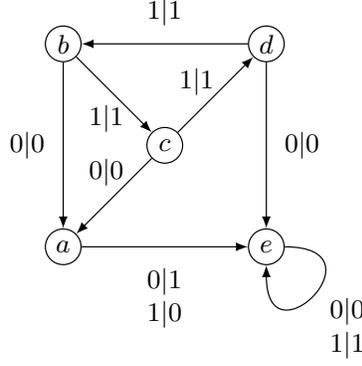
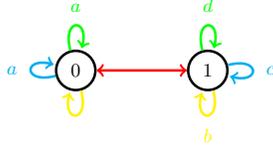Figure 2: The automata generating the first Grigorchuk group.

Figure 3: The Schreier graph on level 1 of the automata generating the first Grigorchuk group.

we know the actual optimal bound but face a more challenging computation challenge. The diagram and Schreier graph of the automaton acting on the third level are drawn in Fig. 6. These Schreier graphs have been extensively studied by Vorobets [54].

- We introduce a close relative of the Grigorchuk groups. It acts on an alphabet with 6 letters. Its stateset is given by

$$a = \langle e, e, e, e, e, e \rangle (1, 2)(3, 4)(5, 6)$$
$$b = \langle e, e, e, e, e, b \rangle (4, 5)$$
$$c = \langle a, e, e, e, e, c \rangle (2, 3)$$
$$d = \langle a, e, e, e, e, d \rangle (2, 3)(4, 5)$$

and its diagram and Schreier graph depicted in Fig. 7. Its introduction is motivated by the similarities with the Schreier graph descriptions the Grigorchuk group in [54] (compare Fig. 6) and the fragmentations of dihedral actions as introduced by Nekrashevych in [45]. Note that the elementary relations $a^2 = b^2 = c^2 = d^2 = bcd = e$ of the Grigorchuk groups are satisfied, however this group does not belong to the class of Grigorchuk groups introduced in [28] because it contains elements of order 6.

- The group introduced by the first author in [16] and whose diagram and Schreier graphs are depicted in Fig. 8 was a motivation for the present work. It is related to Wilson's groups of non-uniform growth [56, 55, 14]. It is generated by an involution and an element of order 3.

- Similarly to the previous example, we introduce a new automaton group which is a close relative of a group of intermediate growth constructed by Bartholdi in relationship with non-uniform growth [4]. Its diagram and Schreier graphs are depicted in Fig. 9. It is generated by three involutions and has exponential activity in the sense of Sidki.

9

- Finally, we provide an example of an automaton group with 9 states acting on an alphabet with 17 letters. Its Schreier graph has the shape of an X but the labellings are inspired by the Schreier graph of the first Grigorchuk group. Its stateset is:

$$a = \langle e,e,e,e,e,e,e,e,e,e,e,e,e,e,e,e,e \rangle (1,2)(6,7)(12,13)(15,16)$$
$$b = \langle e,e,e,e,e,e,e,e,e,e,e,e,e,e,e,e,b \rangle (4,5)(7,8)(1,10)(14,15)$$
$$c = \langle e,e,e,e,e,e,e,e,a,e,e,e,e,e,e,e,c \rangle (4,5)(14,15)$$
$$d = \langle e,e,e,e,e,e,e,e,a,e,e,e,e,e,e,e,d \rangle (7,8)(1,10)$$
$$a' = \langle e,e,e,e,e,e,e,e,e,e,e,e,e,e,e,e,b \rangle (3,4)(8,9)(10,11)(1,14)$$
$$b' = \langle e,e,e,e,b',e,e,e,e,e,e,e,e,e,e,e,e \rangle (2,3)(1,6)(11,12)(16,17)$$
$$c' = \langle e,e,e,e,c',e,e,e,e,e,e,e,a',e,e,e,e \rangle (2,3)(16,17)$$
$$d' = \langle e,e,e,e,d',e,e,e,e,e,e,e,a',e,e,e,e \rangle (1,6)(11,12).$$

It satisfies the relations $a^2 = b^2 = c^2 = d^2 = bcd = a'^2 = b'^2 = c'^2 = d'^2 = b'c'd' = e$.

3 Word norms and growth

3.1 Word lengths and word norms

Let G be a group. A subset S is *generating* (in the sense of semigroups) if for any $g \in G$ there exists $s_1, \ldots, s_n \in S$ such that $g = s_1 \ldots s_n$. In other terms, there is a word in the free semi-group S^* generated by S that equals g once evaluated in G. By convention, we assume that the neutral element e is not in S. A *weight* function on S is a positive function $\pi : S \to \mathbb{R}_{>0}$. A subset S of G is *symmetric* if $S = S^{-1}$ and in this case π is symmetric if $\pi(s^{-1}) = \pi(s)$ for all s in S. Throughout the paper, we consider triples (G, S, π), where G is a group together with a finite generating set S and a weight π on it.

For any $w = s_1 \ldots s_n$ in S^*, we call *length* of w with respect to π the number

$$|w|_\pi := \sum_{i=1}^{n} \pi(s_i).$$

In particular $|s|_\pi = \pi(s)$ for all $s \in S$. By convention the empty sum is zero. For any g in G, we call *norm* of G with respect to (S, π) the number

$$\|g\|_{S,\pi} := \inf \{|w|_\pi : w \in S, w =_G g\}.$$

When π is constant equal to one, we recover the usual word norm with respect to S. In this case, we simply write $\| \cdot \|_S$. The terminology is justified by the

Proposition 3.1. *The function* $\| \cdot \|_{S,\pi} : G \to \mathbb{R}_{\geq 0}$ *satisfies*

(a) $\forall g, h \in G, \|gh\|_{S,\pi} \leq \|g\|_{S,\pi} + \|h\|_{S,\pi}$,

(b) $\|g\|_{S,\pi} = 0$ *if and only if* $g = e$ *is the neutral element.*

Moreover, the infimum is in fact a minimum.

A word w is a *minimal representative* of a group element g if $|w|_\pi = \|g\|_{S,\pi}$. It always exists but may not be unique. Observe that a subword of a minimal word is also minimal by (a). Regarding generators, we always have $|s|_\pi \geq \|s\|_{S,\pi}$, but the converse inequality may not be true. However when we have such a strict inequality, the generator s never appears in minimal representative words, so the generating set is in fact redundant.

Proof. Let $g \neq e$. By assumption there is a word w in S representing g, so $\|g\|_{S,\pi}$ is finite. Since $\pi_0 = \min\{\pi(s)|s \in S\} > 0$, there are only finitely many words in S^* of π-length less than $|w|_\pi$, so the infimum is a minimum. Moreover only the empty word (representing the neutral element) has zero π-length. This prove (b). Finally let w_g and w_h be minimal representative words of g and h, then $w_g w_h$ represents gh, assertion (a) follows. $\square$

This word norm induces a metric $d_{S,\pi}(x,y) := \|x^{-1}y\|_{S,\pi}$ on G. Recall that a metric is a function $d : G \times G \to \mathbb{R}_{\geq 0}$ such that for all $x, y, z \in G$, we have $d(x,z) \leq d(x,y) + d(y,z)$ and that $d(x,y) = 0$ if and only if $x = y$. This metric is left-invariant under the group action in the sense that $d_{S,\pi}(gx, gy) = d_{S,\pi}(x,y)$ for all g, x, y in G. Observe that this metric is symmetric, which means $d_{S,\pi}(x,y) = d_{s,\pi}(y,x)$ for all x, y, if and only if both S and π are symmetric.

When we identify the group G with the vertices of its Cayley graph $\Gamma_{G,S}$, the metric $d_{S,\pi}$ coincides with the metric inherited by the vertex set when we declare that each oriented edge labelled by s has length $\pi(s)$. It implies that the path in $\Gamma_{G,S}$ defined by following edges labelled by the letters of a minimal word w is a geodesic (i.e. shortest) path from the starting point x to the end point $y = xw$.

3.2 Growth functions and growth rate

Definition 3.2. The *ball* of radius ℓ and center x in $(G, d_{S,\pi})$ is the set

$$B_{G,S,\pi}(x,\ell) := \{y \in G : d_{S,\pi}(x,y) \leq \ell\}.$$

The *growth* function of G with respect to (S,π) counts the sizes of the balls

$$\gamma_{G,S,\pi}(\ell) = \#B_{G,S,\pi}(x,\ell).$$

By left-invariance, the size of a ball in a group depends only on the radius ℓ and not on the center x. To fix ideas, one may choose $x = e$.

Let us say that two functions f, g are *equivalent* when there exists $c > 0$ such that $g(\frac{1}{c}\ell) \leq f(\ell) \leq g(c\ell)$ for all $\ell \geq 0$. The next lemma ensures that the equivalence class of $\gamma_{G,S,\pi}(\ell)$ does not depend on S nor on π. This class is called the growth *rate* of G, and denoted by abuse of notation $\gamma_G(\ell)$.

Lemma 3.3. *Let (S,π) and (T,ρ) be two finite weighted generating sets of the group G, then there exists $c > 0$ such that $\gamma_{G,S,\pi}(\ell) \leq \gamma_{G,T,\rho}(c\ell)$ for all ℓ.*

Proof. Let $c := \max\left\{\frac{\|s\|_{T,\rho}}{\|s\|_{S,\pi}} \,\middle|\, s \in S\right\} \in (0, \infty)$. If $g \in B_{G,S,\pi}(\ell)$ then there exists a minimal representative word $w = s_1 \ldots s_n$ in S^* so that

$$\ell \geq \|g\|_{S,\pi} = |w|_\pi = \sum_{i=1}^n \pi(s_i) = \sum_{i=1}^n |s_i|_\pi = \sum_{i=1}^n \|s_i\|_{S,\pi}.$$

A fortiori

$$\|g\|_{T,\rho} \le \sum_{i=1}^{n} \|s_i\|_{T,\rho} \le c \sum_{i=1}^{n} \|s_i\|_{S,\pi} \le c\ell.$$

$\square$

Definition 3.4. A group G has

- *polynomial* growth if $\gamma_G(\ell)$ is bounded above by a polynomial function,

- *exponential* growth if there is $c > 0$ such that $\gamma_G(\ell) \ge \exp(c\ell)$ for all ℓ,

- *intermediate* growth if $\gamma_G(\ell)$ is greater than any polynomial function and lesser than any exponential function:

$$\forall d \in \mathbb{Z}, \alpha \in \mathbb{R}_{>0}, \exists N, \forall \ell \ge N, \ell^d \le \gamma_G(\ell) \le \exp(\alpha\ell).$$

4 Essentially geodesic generating sets (eggs)

This section is devoted to the key notion of essentially geodesic generating sets, in short eggs, of a triple (G, S, π).

4.1 Definition and examples

Definition 4.1. A finite subset Δ of S^* is an *essentially geodesic generating* set (in short, an egg) for (G, S, π) if there is a finite subset F of S^* such that for any $g \in G$, there exist $\delta_1, \dots, \delta_n \in \Delta$ and $h \in F$ such that the word $\delta_1 \dots \delta_n h$ is a minimal representative of g with respect to (S, π).

Clearly, the words δ_i of the definition have to be minimal themselves. For example, S itself is an egg, with F the empty set. Let us give more examples.

Let $\Omega \subset G$. Define its boundary as $\partial\Omega := \{g \in \Omega : \exists s \in S, gs \notin \Omega\}$. Denote $\mathrm{Int}(\Omega) := \Omega \setminus \partial\Omega$ the interior of Ω. In the Cayley graph, vertices in the boundary of Ω are precisely those with at least one neighbor outside Ω.

Lemma 4.2. *Assume $\Omega \subset G$ is finite and $e \in \mathrm{Int}(\Omega)$. Then a set Δ of minimal representative words of the boundary $\partial\Omega$ is an egg of G.*

Proof. Given $g \in G$, let $w = s_1 \dots s_n$ be a minimal representative word for (S, π). Starting from e and following the edges labelled by the letters of w provides a geodesic path to g in the Cayley graph. If $g \notin \mathrm{Int}(\Omega)$, this path must cross $\partial\Omega$. This means there is a non-trivial prefix (necessarily minimal) w_1 of w representing an element in $\partial\Omega$. By assumption, there is $\delta_1 \in \Delta$ of the same length as w_1 such that $\delta_1 =_G w_1$. Then $\delta_1^{-1}g$ is an element in G at distance from e less than $d_{S,\pi}(e, g) - \pi_0$, where $\pi_0 := \min\{\pi(s)|s \in S\} > 0$. Arguing by induction, we can write $g = \delta_1 \dots \delta_k h$ with $\delta_i \in \Delta$ and $h \in \mathrm{Int}(\Omega) =: F$ which is finite. $\square$

Of course not all eggs are of this form, because any set of words containing an egg is still an egg. But there are also eggs containing no such $\partial\Omega$ as seen by the following.

Example 4.3. Let $S = \{a^{\pm 1}, b^{\pm 1}\}$ be the usual generating set of $\mathbb{Z}^2$ where $a = (1, 0), b = (0, 1)$. Then for integers $p, q > 0$, the set $\Delta = \{a^{\pm p}, b^{\pm q}\}$ is an egg, because any $g \in \mathbb{Z}^2$ can be written minimally $g = a^{\lambda p}b^{\mu q}a^r b^s$ with λ, μ, r, s integers and $|r| < p$, $|s| < q$. However the minimal word $(ab)^k$ has no prefix in Δ.

4.2 Stability under quotients

Let $\widetilde{G}$ be a group together with a quotient group G. Denote by $q : \widetilde{G} \to G$ the quotient map. We assume that $\widetilde{S}$ is a finite generating set of $\widetilde{G}$ and that $q|_{\widetilde{S}}$ is injective, inducing a bijection between $\widetilde{S}$ and $S := q(\widetilde{S})$. Given a word $\widetilde{w} = \widetilde{s_1} \ldots \widetilde{s_n}$ in $\widetilde{S}$, we systematically write $w = s_1 \ldots s_n$ instead of $q(\widetilde{w}) = q(\widetilde{s_1}) \ldots q(\widetilde{s_n})$ the corresponding word in S. Clearly, S is a generating set of G.

Let $\pi : \widetilde{S} \to \mathbb{R}_{>0}$ be a weight function, it naturally induces a weight function on S still denoted π. For any $\widetilde{g}$ in $\widetilde{G}$ we observe that $\|q(\widetilde{g})\|_{S,\pi} \leq \|\widetilde{g}\|_{\widetilde{S},\pi}$. Indeed, any word in $\widetilde{S}$ representing $\widetilde{g}$ corresponds to a word in S of the same π-length representing $q(\widetilde{g})$, so the infimum defining the left-hand side is taken over a bigger set of words.

Lemma 4.4. *In the setting above, assume that $\widetilde{\Delta}$ is an egg for $(\widetilde{G}, \widetilde{S}, \pi)$. Then $\Delta := q(\widetilde{\Delta})$ is an egg for (G, S, π).*

Proof. Let g belong to G and w be a minimal representative word of g in S. Then $\widetilde{w}$ is a minimal word in $\widetilde{S}$ (by the observation before the lemma). As $\widetilde{\Delta}$ is an egg, we have $\widetilde{w} =_G \widetilde{\delta_1} \ldots \widetilde{\delta_n} \widetilde{h} =: \widetilde{w_1}$, a minimal word with $\widetilde{\delta_i}$ in $\widetilde{\Delta}$ and $\widetilde{h}$ in some finite set $\widetilde{F}$ of words in $\widetilde{S}$. A fortiori, $w_1 = \delta_1 \ldots \delta_n h$ is the required representative word of g. $\square$

Example 4.5. For G the first Grigorchuk group as depicted in Fig. 2, it is natural to consider $\widetilde{G} = \langle a, b, c, d | a^2 = b^2 = c^2 = d^2 = bcd = e \rangle$, which is isomorphic to the free product $\mathbb{Z}/2\mathbb{Z} * (\mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z})$.

4.3 Stability of eggs under change of weights

Consider an egg Δ for (G, S, π). We want to know under what condition the set Δ is still an egg for (G, S, π'). Clearly this is the case if among words in S, minimality for $\| \cdot \|_{S,\pi}$ is equivalent to minimality for $\| \cdot \|_{S,\pi'}$.

Lemma 4.6. *Let $G := \mathbb{F}(X)$ be the free group with finite free generating set X. Let $S = X \cup X^{-1}$ and $\pi : S \to \mathbb{R}_{>0}$ be a weight. A word representative is minimal for $\| \cdot \|_{S,\pi}$ if and only if it is minimal for the uniform word norm $\| \cdot \|_S$.*

In particular, a set Δ is an egg for $\| \cdot \|_{S,\pi}$ if and only if it is an egg for the uniform word norm $\| \cdot \|_S$.

Proof. As the Cayley graph is a tree, minimal representatives are unique and correspond to non-backtracking paths. Therefore they coincide for both norms. $\square$

We can obtain a similar stability result in the context of free products of finite groups, but the weights need to satisfy some condition.

Lemma 4.7. *Let $G = *_{i \in I} G_i$ be a free product of finitely many finite groups. Let $S = \cup_{i \in I} G_i \setminus \{e\}$ and $\pi : S \to \mathbb{R}_{>0}$ be a weight function such that*

$$\forall i \in I, \forall s, s' \in G_i \setminus \{e\}, \quad \pi(ss') < \pi(s) + \pi(s'). \tag{4}$$

Then the following are equivalent:

(i) *a word $w = s_1 \ldots s_n$ is minimal for $\| \cdot \|_{S,\pi}$,*

(ii) *the word w is freely reduced, i.e. for all $1 \leq k \leq n - 1$, the generators s_k and s_{k+1} belong to different factor groups G_i.*

13

A weight function satisfying condition (4) is called *triangular*. This condition is essentially necessary because if there are $s, s' \in G_i$ with $s_0 = ss'$ and $\pi(s_0) > \pi(s) + \pi(s')$ then s_0 would be a freely reduced word not minimal for $\| \cdot \|_{S,\pi}$.

As the usual word norm satisfies condition (4), minimality for $\| \cdot \|_S$ and for $\| \cdot \|_{\pi,S}$ are equivalent. We deduce the:

Corollary 4.8. *Let $G = *_{i \in I} G_i$ and S, π be as in Lemma 4.7. A set Δ is an egg for $\| \cdot \|_{S,\pi}$ if and only if it is an egg for the usual word norm.*

Proof of Lemma 4.7. If a word is minimal, then successive letters must belong to different factors, otherwise using (4) we would replace the successive letters s_k and s_{k+1} by their product, thus obtaining the same group element but with a representative of shorter π-length.

Conversely assume that w is freely reduced, but not minimal. Then there exists $w' = s'_1 \ldots s'_m =_G w$ a minimal representative word for $\| \cdot \|_{S,\pi}$. By the first part of the proof, we can assume w' is freely reduced. Now let $k_0 := \min\{k : s'_k \neq s_k\}$, then $s_{k_0} \ldots s_n =_G s'_{k_0} \ldots s'_m$ which implies $s'^{-1}_m \ldots s'^{-1}_{k_0} s_{k_0} \ldots s_n =_G e$. We obtain a freely reduced representative word of the identity. As G is a free product, this is a contradiction unless w and w' are the same word. $\square$

The context of free products of finite groups is restrictive. However, it seems delicate to obtain more general results about stability of eggs, as shown by the following:

Example 4.9. Consider the group $\mathbb{Z}^2$ with $S = \{(\pm 1, 0), (0, \pm 1), \pm(1,1)\}$ and two different weigths $\pi_1(\pm 1, 0) = \pi_1(0, \pm 1) = 1$, $\pi_1(\pm(1,1)) = 2$ and $\pi_2(\pm 1, 0) = \pi_2(0, \pm 1) = 1$, $\pi_2(\pm(1,1)) = \lambda \in (0,2)$. Then the norm $\| \cdot \|_{S,\pi_1}$ coincides with the usual word metric on the grid so that $\Delta := \{(\pm 1, 0), (0, \pm 1)\}$ is an egg. However Δ is no longer an egg for $\| \cdot \|_{S,\pi_2}$ because $\|(n,n)\|_{S,\pi_2} = \lambda n$ is much smaller than the distance between e and any fixed neighborhood of (n,n) in the usual word norm which is at least $2n - c$.

5 The strong contraction method for subexponential growth

The following theorem will be our key tool to obtain upper bounds on growth exponents. The general strategy is due to Grigorchuk [27, 28], the use of weights appeared in [3] and the following precise statement is taken from [42]. This method is so far the only known method to obtain non-trivial upper bounds on growth exponents.

Theorem 5.1 (Muchnik-Pak [42]). *Given a triple (G, S, π), assume that there exist $\eta \in [0,1]$ and $c_0 > 0$ together with embeddings for all ℓ*

$$B_{G,S,\pi}(\ell) \hookrightarrow \bigcup_{\ell_1 + \cdots + \ell_d \leq \eta\ell + c_0} K \times \prod_{i=1}^{d} B_{G,S,\pi}(\ell_i)$$

where K is a fixed finite set. Then there is a constant $c > 0$ such that $\gamma_{G,S,\pi}(\ell) \leq e^{c\ell^\alpha}$ with $\alpha = \frac{\log(d)}{\log(d) - \log(\eta)}$.

Note that the assumptions need to be satisfied by one weighted generating set and the conclusion then applies to the growth rate of G, hence for all weighted generating set. However in practice, the constants η, c_0, c and the finite set K depend on (S, π). The key point in the sequel will be to find good generating sets and weights to lower η and thus α. Of course, this theorem is relevant only in the case $\eta < 1$. To apply it, we will systematically use the following:

Proposition 5.2. *Let G be a self-similar group with $\psi : G \hookrightarrow G \wr_{\Sigma} \mathrm{Sym}(\Sigma)$. Assume there is a weighted norm $\|\cdot\|_{S,\pi}$ with an egg Δ and a constant $\eta \in [0,1]$ such that*

$$\forall \delta \in \Delta, \quad \sum_{x \in \Sigma} \|\delta_x\|_{S,\pi} \leq \eta \|\delta\|_{S,\pi} \tag{$\star$}$$

where $\psi(\delta) = \langle \delta_{x_1}, \ldots, \delta_{x_d} \rangle \sigma$ is the self-similar image of δ. Then the assumptions of Theorem 5.1 are satisfied with the same η.

We call the minimal such $\eta = \eta_{S,\pi}(\Delta)$ the *contraction coefficient* of the egg Δ. In order to get upper bounds on growth, we will construct eggs with small contraction coefficient.

Proof. Take $K = \mathrm{Sym}(\Sigma)$. The permutational wreath products defines an embedding $G \hookrightarrow \mathrm{Sym}(\Sigma) \times G^{\Sigma}$ by declaring that the image of $g = \langle g_{x_1}, \ldots, g_{x_d} \rangle \sigma$ is the $d+1$-tuple $(\sigma, g_{x_1}, \ldots, g_{x_d})$. It is sufficient to prove that the image of each element g in $B_{G,S,\pi}(\ell)$ is in the desired union, i.e. that $\sum_{x \in \Sigma} \|g_x\|_{S,\pi} \leq \eta \ell + c_0$.

As Δ is an egg, we can write $g = \delta^1 \ldots \delta^n h$ minimal representative with $\delta^i \in \Delta$ and $h \in F$ (we use superscripts here only to avoid confusion with subscripts corresponding to sections). Then

$$
\begin{aligned}
g &= \left(\prod_{i=1}^{n} \left\langle \delta^i_{x_1}, \ldots, \delta^i_{x_d} \right\rangle \sigma^i \right) \langle h_{x_1} \ldots, h_{x_d} \rangle \sigma' \\
&= \left(\prod_{i=1}^{n} \left\langle \delta^i_{x_1}, \ldots, \delta^i_{x_d} \right\rangle^{\tau_{i-1}} \right) \langle h_{x_1}, \ldots, h_{x_d} \rangle^{\tau_n} \tau_n \sigma', \quad \text{where} \quad \tau_i = \sigma^1 \ldots \sigma^i, \\
&= \left(\prod_{i=1}^{n} \left\langle \delta^i_{\tau_{i-1}^{-1}.x_1}, \ldots, \delta^i_{\tau_{i-1}^{-1}.x_d} \right\rangle \right) \left\langle h_{\tau_n^{-1}.x_1}, \ldots, h_{\tau_n^{-1}.x_d} \right\rangle \tau_n \sigma' \\
&= \left\langle \left(\prod_{i=1}^{n} \delta^i_{\tau_{i-1}^{-1}.x_1} \right) h_{\tau_n^{-1}.x_1}, \ldots, \left(\prod_{i=1}^{n} \delta^i_{\tau_{i-1}^{-1}.x_d} \right) h_{\tau_n^{-1}.x_d} \right\rangle \tau_n \sigma'.
\end{aligned}
$$

By injectivity of the embedding, we have $g_x = \left(\prod_{i=1}^{n} \delta^i_{\tau_{i-1}^{-1}.x} \right) h_{\tau_n^{-1}.x}$ for each section x and $\sigma = \tau_n \sigma'$. Therefore

$$
\begin{aligned}
\sum_{t=1}^{d} \|g_t\|_{S,\pi} &\leq \sum_{t=1}^{d} \sum_{i=1}^{n} \left\| \delta^i_{\tau_{i-1}^{-1}.t} \right\|_{S,\pi} + \left\| h_{\tau_n^{-1}.t} \right\|_{S,\pi} \\
&= \sum_{i=1}^{n} \sum_{t=1}^{d} \left\| \delta^i_t \right\|_{S,\pi} + \sum_{t=1}^{d} \|h_t\|_{S,\pi} \leq \sum_{i=1}^{n} \eta \left\| \delta^i \right\|_{S,\pi} + c_0 \leq \eta \|g\|_{S,\pi} + c_0 \leq \eta \ell + c_0.
\end{aligned}
$$

Note that c_0 is at most d times the maximal length of a section of an element in F. $\qquad\square$

6   Semi-algorithms computing upper bounds

In order to apply the contraction criterion of the previous section and obtain an upper exponent α as small as possible, we have to find an egg Δ with minimal coefficient η for some norm $\|\cdot\|_{S,\pi}$. More precisely, we want to estimate

$$\bar{\eta}(G) := \inf_{(S,\pi)} \inf_{\Delta} \max_{w \in \Delta} \bar{\eta}_{S,\pi}(w), \quad \text{where} \quad \bar{\eta}_{S,\pi}(w) = \frac{\sum_{x \in \Sigma} \|w_x\|_{S,\pi}}{\|w\|_{S,\pi}}, \tag{5}$$

where the first infimum is over all weighted generating sets (S, π) and the second is over all eggs Δ for $\| \cdot \|_{S,\pi}$. However there is a last technical issue that it is difficult to estimate the actual norm $\|w\|_{S,\pi}$ of a word (or rather of the element represented by this word), and much easier to estimate the length $|w|_\pi$. So in practice we will evaluate

$$\eta(G) := \inf_{(S,\pi)} \inf_\Delta \max_{w\in\Delta} \eta_{S,\pi}(w), \quad \text{where} \quad \eta_{S,\pi}(w) = \frac{\sum_{x\in\Sigma} |w_x|_\pi}{|w|_\pi}, \tag{6}$$

This is sufficient because of the

Fact 6.1. $\bar{\eta}(G) \le \eta(G)$.

Proof. Recall that we always have $\|w\|_{S,\pi} \le |w|_\pi$. For each w in a fixed egg Δ, we have

$$\bar{\eta}_{S,\pi}(w) \le \frac{\sum_{x\in\Sigma} |w_x|_\pi}{\|w\|_{S,\pi}} = \frac{\sum_{x\in\Sigma} |w_x|_\pi}{|w|_\pi} = \eta_{S,\pi}(w),$$

except if $\|w\|_{S,\pi} < |w|_\pi$ which means that w is not minimal for (S, π). But in the latter case we can safely remove w from Δ and still obtain an egg $\Delta' = \Delta \setminus \{w\}$. $\qquad\square$

Finally if we have (6), we deduce (5). Then Theorem 5.1 and Proposition 5.2 give

$$\gamma_G(\ell) \le e^{c\ell^\alpha}, \quad \text{where} \quad \alpha = \frac{\log(d)}{\log(d) - \log(\eta(G))} \text{ and } d = \#\Sigma.$$

Automata groups come naturally with a preferred generating set (in bijection with the stateset). For simplicity, we have chosen to use this particular one and not change it.

6.1 A semi-algorithm with fixed weights

Let (G, S, π) be fixed. We explain here how to obtain upper bounds on

$$\eta(G, S, \pi) := \inf_\Delta \eta_{S,\pi}(\Delta), \quad \text{where} \quad \eta_{S,\pi}(\Delta) = \max_{w\in\Delta} \eta_{S,\pi}(w).$$

The first issue is to ensure the existence of an egg with $\eta_{S,\pi}(\Delta) < 1$.

6.1.1 A procedure to obtain eggs.

Let $P(w)$ be a property of words in S, and assume that we are able to determine algorithmically whether $P(w)$ is true or not for a given w. We describe a procedure aiming at producing an egg Δ of (G, S, π) such that $P(w)$ is true for all w in Δ.

At time one, we set $L = S$ as a list. (We recall that $e \notin S$.) At each time, we take the first word w in the list, we remove it from L and do

(a) if P(w) is true, then add w to the set Δ,

(b) if P(w) is false, then for each $s \in S$

  (i) if we detect that $\|ws\|_\pi < \|w\|_\pi + \|s\|_\pi$, do nothing,

  (ii) elif we detect u in L such that $ws =_G u$, do nothing,

  (iii) otherwise we add ws at the end of L.

The procedure stops once L is empty.

Lemma 6.2. *If the procedure above stops in finite time, then the resulting set Δ is an egg of (G, S, π) such that $P(w)$ is true for all w in Δ.*

Proof. Let g be an element of G with a representative $w = s_1 \ldots s_m$ minimal for $\|\cdot\|_{S,\pi}$. Arguing by induction, it is sufficient to prove that if $\|g\|_{S,\pi}$ is large enough, there is a prefix of w which is equal in G to a word δ_1 of Δ.

Assume by contradiction that no prefix of w is equal in G to a word in Δ. Then we claim that for each $k \geq 1$, there exists a word w_k that was in L at some time during the algorithm such that $g =_G w_k s_{k+1} \ldots s_m$.

Indeed, this is obvious for $k = 1$. Assume this is true for k and consider $w_k s_{k+1}$. This word is not in Δ by assumption, so case (a) does not apply. It is of length $\|w_k\|_{S,\pi} + \|s_{k+1}\|_{S,\pi}$ by minimality of w, so case (b)(i) does not apply. Therefore either case (b)(ii) $w_k s_{k+1}$ equals in G to some $u =: w_{k+1}$ that was in L at the same time, or case (b)(iii) $w_k s_{k+1} =: w_{k+1}$ was added to L.

If the algorithm stopped in finite time, there has been finitely many elements in L and we get a contradiction as soon as $|w_k|_\pi$ is bigger than all their π-lengths. $\qquad\square$

In fact the lemma still holds if we do not try to detect (b)(i) or (b)(ii). However the resulting egg Δ (if it exists) would be bigger and the time of computation would be larger.

6.1.2 The target semi-algorithm.

Given an automaton group G with a fixed weighted generating system (S, π), we use a semi-algorithm, presented in Fig. 4, to obtain an upper bound on $\eta(G, S, \pi)$. This semi-algorithm implements the procedure described above with the following property.

For a target coefficient $\eta_{\text{tar}} \in (0, 1]$, we say a word w satisfies property $P_{\eta_{\text{tar}}}(w)$ if it satisfies condition $(\star)$ of Proposition 5.2 for $\eta = \eta_{\text{tar}}$, i.e.

$$\eta_{S,\pi}(w) = \frac{\sum_{x \in \Sigma} |w_x|_\pi}{|w|_\pi} \leq \eta_{\text{tar}}$$

Lemma 6.2 guarantees that if the semi-algorithm stops, it furnishes an egg for $\|\cdot\|_{S,\pi}$ all of which element have property $P_{\eta_{\text{tar}}}(w)$.

In order to try to detect (b)(i) when $\|ws\|_{S,\pi} < \|w\|_{S,\pi} + \|s\|_{S,\pi}$ we consider an auxiliary group $\widetilde{G}$ of which G is a quotient (so minimality in G implies minimality in $\widetilde{G}$ but not conversely). Then $\|\cdot\|_{\widetilde{G},\widetilde{S}}$ just denotes the usual word norm with respect to $\widetilde{S}$ as in Section 4.2. In practice, this auxiliary group will be either a free group or a finite free product of free groups.

We also use an auxiliary list L_{aux}. All the elements of this auxiliary list have the same $\|\cdot\|_{\widetilde{G},\widetilde{S}}$-length, and the evolution of the size of this list (with length of words) gives a good heuristical information whether the procedure is about to succeed or fail.

Note that according to (b)(ii), we add the word ws to the auxiliary list L_{aux} only if it represents a new group element. This is what we mean by $\cup_G$. We are aware that this uncontrolled choice of representative word (a priori) impacts the final bound but it would be substantially more difficult to take this into account and seek the best representative.

Finally, the egg obtained obviously satisfies that $\eta_{S,\pi}(w) \leq \max_{w \in \Delta} \eta_{S,\pi}(w) =: \eta_{\max}$ for all w in Δ, which we can record along the run. We conclude

$$\eta(G, S, \pi) := \inf_\Delta \eta_{S,\pi}(\Delta) \leq \eta_{\max}.$$

This algorithm was implemented in **GAP** by the function `IsSubExp_rec`. It is also performed by `IsSubExp_fast` which is faster (it avoids functions calling). When the weights are uniform, the function `IsSubExp_fast_uni` is even faster. The code is given at https://www.irif.fr/~godin/automatongrowth.html.

Data: An automaton group G, S, π and a target coefficient $\eta_{\text{tar}} \in (0, 1]$.
Result: An egg Δ and a coefficent $\eta_{\max} \leq \eta_{\text{tar}}$ such that $\eta_{S,\pi}(w) \leq \eta_{\max}$ for all $w \in \Delta$

$L \leftarrow S$;
$\eta_{\max} \leftarrow 0$;
while $L \neq \emptyset$ **do**
    $L_{\text{aux}} \leftarrow \emptyset$;
    for $w \in L$ **do**
        if $\eta(w) \leq \eta_{\text{tar}}$ **then**
            $\Delta \leftarrow \Delta \cup \{w\}$;
            $\eta_{\max} \leftarrow \max\left(\eta_{\max}, \eta_{S,\pi}(w)\right)$;
        else
            for $s \in S$ **do**
                if $|ws|_{\widetilde{G},\widetilde{S}} = |w|_{\widetilde{G},\widetilde{S}} + |s|_{\widetilde{G},\widetilde{S}}$ **then**
                    $L_{\text{aux}} \leftarrow L_{\text{aux}} \cup_G \{ws\}$;
                end
            end
        end
    end
    $L \leftarrow L_{\text{aux}}$;
end

Figure 4: The target semi-algorithm of Section 6.1.2, implemented as `IsSubExp_rec` in GAP.

There the list Δ is called *shell* and the list L is called the *yolk*. The algorithm stops once the yolk list is empty. While running the algorithm, we record their sizes which give a good heuristical indication whether the algorithm is likely to finish or not. Moreover in order to avoid endless computation, we input a *radius*, which is the maximal length (uniform in S) of words allowed in the shell list Δ and the yolk list L.

6.2 Improving weights for a fixed egg

Once we obtain an egg, we wish to modify the weights in order to minimize $\eta_{S,\pi}(\Delta)$. The difficulty is twofold. First by construction, the fact that a set is an egg depends a priori strongly on the weights chosen. Second we need to actually minimize a function of the form $\eta_{S,\pi}(\Delta) = \max_{w \in \Delta} \eta_{S,\pi}(w)$, where $\eta_{S,\pi}(w)$ is a rational function of $\pi \in (0, 1]^S$.

The choice of auxiliary group $\widetilde{G}$ permits to bypass the first difficulty. Indeed, we choose $\widetilde{G}$ to be either a free group or a finite free product of finite groups. In the first case, we impose no restriction on π while in the second case we impose that π satisfies the triangular conditions (4) of Lemma 4.7. For such weights in such groups, minimality of representative words is invariant, so the detections (b)(i) when we perform the target semi-algorithm do not depend on the choice of weight. This guarantees that we can change weights freely as long as we respect the triangular conditions.

There remains the second task to evaluate the following function of π:

$$\eta(\Delta) = \inf_{\pi \in \Omega} \max_{w \in \Delta} \eta_\pi(w), \quad \text{where} \quad \eta_\pi(w) = \frac{\sum_{x \in \Sigma} |w_x|_\pi}{|w|_\pi}.$$

18

The domain Ω is a subset of $[0,1]^S$ bounded by the triangular conditions (4) and the functions $\eta_\pi(w)$ are rational functions of π. Indeed for a word w in S and an element $s \in$ S, let us denote by $N_s(w)$ the number of occurences of the letter s in w. Then $|w|_\pi = \sum_{s \in S} N_s(w)\pi(s)$. It follows that

$$\eta_\pi(w) = \frac{\sum_{s \in S} \sum_{x \in \Sigma} N_s(w_x)\pi(s)}{\sum_{s \in S} N_s(w)\pi(s)}.$$

When we run the target semi-algorithm, we register the coefficients $N_s(w)$ and $N_s(w_x)$ for $s \in S$ and $x \in \Sigma$.

To concretely estimate $\eta_\pi(\Delta)$, we use a numerical algorithm `wmo` based on a generalized gradient method. `wmo` comes from recursive multi-level solution of a second order dynamical system with embeded ad-hoc inertia effects to escape from local minima. The algorithm is described in [40, 41]. In implementation, the overall `GAP`-program calls for the application `wmo` providing it with the coefficients above and the triangular constraints. The algorithm `wmo` returns the estimated minimum together with the estimated optimal weights.

It seems very challenging to make exact optimization here because the function to minimize is not convex and has several local minima. It also appeared experimentally on some sets of data that very different weights π can lead to values extremely close to the observed minimum.

6.3 Mixed strategies

The minimization problem (6) is intrinsically two dimensionnal, since we have to find both an optimal weight π and an optimal egg Δ. (Recall that in order to simplify the problem, we chose to fix once and for all the generating set S.) The algorithms described previously are essentially one dimensionnal, and they have to be coupled to obtain good results. For this, we implemented three `GAP`-functions described at https://www.irif.fr/~godin/automatongrowth.html.

- The function `IsSubExp_opt` takes as input a given weight π and a target η_{tar}, it searches an egg according to the target semi-algorithm. Once it obtains an egg Δ_1 it calls `wmo` in order to find the weights π_1 optimizing $\eta(\Delta_1)$. This algorithm can be looped by searching for a new egg Δ_2 using the weights π_1 for a new target $\eta_{\mathrm{tar},2}$.

- Given an update parameter, the function `IsSubExp_ovi` takes as input a given weight π and a target η_{tar}, it searches an egg according to the target semi-algorithm, but everytime the yolk list contains words of length multiple of the update parameter, it applies the weight-optimization algorithm `wmo` to the whole set of words $\Delta \cup L$, and updates the weight vector according to the output. It stops once it reaches the target.

- The function `IsSubExp_loop` permits the user to control the strategy. At each step the function asks for an instruction either of exploration (applying the main loop of the target algorithm of Fig. 4 to a chosen subset of $\Delta \cup L$) or of weight optimization (using `wmo` over a chosen subset of $\Delta \cup L$).

7 Data and comments

7.1 Test of implemented functions

We use the first Grigorchuk group in order to test our functions. We consider it together with its usual Mealy automaton, but also with the associated automata of the actions on the second and third levels. In these disguised descriptions, it is much harder to obtain good numerical upper bounds.

7.1.1 Test of `IsSubExp_rec`

In view of [3, 22], we know the optimal weights are $[.305061, .34747, .223839, .123631]$ for $S = \{a, b, c, d\}$ and the optimal contraction coefficients η are $\eta_{\text{opt}} = .8106$, $\eta_{\text{opt}}^2 = .6572$ and $\eta_{\text{opt}}^3 = .5327$ for level 1, level 2 and level 3 respectively.

Indeed, iterating the condition of Theorem 5.1 ensures that

$$B_{G,S,\pi}(\ell) \hookrightarrow \bigcup_{\ell_1 + \cdots + \ell_{d^2} \leq \eta^2 \ell + c_1} K^{d+1} \times \prod_{i=1}^{d^2} B_{G,S,\pi}(\ell_i),$$

so if η is valid for the first level, then η^2 is valid for the second level. By induction η^k is valid also for the kth level. However, a computationnal issue is the increase of the additive constant $c_1 = \eta c_0 + (d+1)c_0$. This additive constant is unessential in theory, but in practice it has to be taken care of by an over approximation of η in the ratios ($\star$) of Proposition 5.2. This explains why the radius and size of eggs blow up as we tighten our precision on the target. We explored numerically this dependance on the target for the optimal weights in Table 4.

level	target	radius	egg size	η	α
1	.90	2	4	.8106	.7675
	.9	8	12	.8857	.9195
	.75	14	73	.7497	.8280
2	.70	25	1360	.6997	.7952
	.69	33	5947	.6900	.7889
	.68	45	93855	.6800	.7824
	.9	9	32	.8954	.9496
	.80	11	52	.7980	.9022
	.70	17	154	.6999	.8536
3	.65	23	427	.6493	.8281
	.60	36	3715	.5999	.8028
	.59	40	7499	.5900	.7977
	.58	46	19616	.5800	.7924

Table 4: Testing the function `IsSubExp_rec` on the first Grigorchuk group with optimal weights.

We also studied the dependance on the weights, using perturbations of the (normalized) optimal weights by vectors $\lambda[1, -1, 1, -1]$ for $\lambda = .02$ and $\lambda = .005$. The data are given in Table 5.

It appears that in some situations the non-optimal weights provide (very slightly) smaller eggs, but as we tighten our target precision, this is no longer the case. For this reason, it will be difficult to detect whether weights are optimal or not.

7.1.2 Test of `IsSubExp_ovi`

Given a target, this function calls `wmo` in order to optimize weights once an egg for the initial weights is obtained. We would like to use it in order to find suitable weights for new automata groups where optimal weights are unknown. Data obtained for the Grigorchuk group is described in Table 6

A first flaw is that the existence of such an egg depends on the weights. For instance for uniform weights on the Grigorchuk group at first or second level, no egg is obtain so the algorithm does not end. The reason is that words of the form $(ab)^k$ are not contracting. An easy way to bypass this

weights on first Grigorchuk group	level	target	radius	egg size	η	α
	1	.90	2	4	.8719	.8349
	2	.75	17	88	.7471	.8263
[.325061, .32747, .243839, .103631]		.70	89	28645	.7000	.7954
optimal $\pm$ .02		.65	25	418	.6497	.8283
	3	.60	40	4145	.6000	.8028
		.59	53	9672	.5900	.7977
		.58	69	33001	.5800	.7925
	1	.90	2	4	.8261	.7840
	2	.75	15	76	.7419	.8229
[.310061, .34247, .228839, .118631]		.70	29	1595	.7000	.7954
optimal $\pm$ .005		.65	23	427	.6499	.8284
	3	.60	35	3576	.6000	.8028
		.59	40	7477	.5900	.7977
		.58	50	20744	.5800	.7925

Table 5: Dependance of radius and size of eggs on the weights – compare also with Table 4.

difficulty is to break the symmetry and use almost uniform weights. However not all symmetry breaking works.

A second flaw appears for instance on level two with almost uniform initial weights. The output weights turn out to be a permutation of the optimal weights and when we plug in these weights with a tighter target, the algorithm does not end. We reached a local minimum of our optimization problem and this function cannot exit it.

When given aa initial weight close to the optimal value (say [.30, .35, .20, .10]), the function outputs a slightly better weight. But as seen on Table 6 the evolution of the weights and contraction coefficient η is very slow and it is not clear whether they would often converge to the optimal vector or to unsatisfying local minima.

7.1.3   Test of `IsSubExp_opt`

Given a fixed target, this function looks for an egg, but every update round it applies the optimizer `wmo` to the whole set of words $\Delta \cup L$, then it continues using the new weights until next update. Experimental data are given in Table 7.

At the first level, the results and output weights are very close to optimal, but it may be due to the simplicity of the problem. At the second level, they are still close to optimal but the obtained eggs are substantially bigger. Again, the relative simplicity of the problem might account for this success.

At the third level, it is much harder to obtain satisfying output vectors. The best results are obtained with tighter targets and sparser updates, but at the expense of longer computations.

Note that on level 3, for target .65 and update 4, at round 28 the use of `wmo` provided the following vector [.2959, .3521, .2705, .0817], much closer to the optimal vector than the final output – see Table 7.

level	loop	target	initial weights	radius	egg size	η	α	output weights
1	1	.9999	uniform	does not end				
	1	.9999	[1.,.99,.98,.97]	2	4	.8119	.7689	[.3067,.3467,.2232,.1236]
	1	.9999	[1.,.97,.98,.99]	does not end				
2	1	.9999	uniform	does not end				
	1	.9999	[1.,.99,.98,.97]	5	8	.8107	.8686	[.3052,.3474,.1237,.2239]
	2	.8106	from loop 1	does not end				
	1	.80	[.30,.35,.20,.10]	10	30	.7901	.8548	[.2851,.3575,.2251,.1325]
	2	.75	from loop 1	14	93	.7463	.8258	[.2866,.3568,.2257,.1311]
	3	.70	from loop 2	30	1700	.6999	.7953	[.2865,.3568,.2258,.1312]
3	1	.9999	uniform	9	32	.8508	.9279	[.3164,.3419,.1710,.1710]
	2	.85	from loop 1	10	36	.7992	.9027	[.3563,.3219,.1917,.1303]
	3	.70	from loop 2	17	147	.6910	.8491	[.3720,.3141,.1932,.1209]
	4	.65	from loop 3	23	361	.6491	.8280	[.3736,.3132,.1934,.1199]
	5	.63	from loop 4	30	810	.6281	.8173	[.3696,.3153,.1956,.1198]
	6	.61	from loop 5	42	2155	.6099	.8079	[.3687,.3157,.1968,.1190]
	7	.60	from loop 6	48	3964	.6000	.8028	[.3682,.3159,.1969,.1191]
	1	.7	[.40,.30,.10,.20]	40	315	.6971	.8522	[.4024,.2988,.1035,.1955]
	2	.67	from loop 1	166	2313	.6672	.8371	[.4097,.2952,.1107,.1846]
	3	.66	from loop 2	114	2636	.6587	.8328	[.4023,.2989,.1178,.1812]
	4	.65	from loop 3	234	4854	.6492	.8280	[.3998,.3001,.1195,.1808]
	1	.70	[.30,.35,.20,.10]	16	154	.6966	.8519	[.2943,.3529,.2286,.1244]
	2	.65	from loop 1	23	535	.6497	.8283	[.2966,.3517,.2275,.1243]

Table 6: Testing the function IsSubExp_ovi on the first Grigorchuk group.

level	target	update	radius	egg size	η	α	output weights
1	.90	2	2	4	.8113	.7683	[.3052,.3475,.2243,.1232]
		4	5	8	.8107	.7676	[.3053,.3474,.2238,.1236]
		10	11	64	.8136	.7707	[.3118,.3442,.2219,.1224]
2	.90	2	2	4	.8113	.8690	[.3100,.3452,.1227,.2224]
		4	4	8	.8121	.8695	[.3072,.3465,.1229,.2236]
		10	10	32	.8124	.8697	[.3089,.3456,.1232,.2534]
	.75	2	16	284	.7467	.8260	[.3071,.3465,.1912,.1554]
		4	18	315	.7487	.8273	[.3129,.3436,.1922,.1515]
		10	24	536	.7499	.8281	[.3296,.3353,.1822,.1532]
	.72	4	32	805	.7166	.8063	[.3069,.3466,.2068,.1399]
		10	36	1199	.7198	.8083	[.3051,.3475,.2086,.1390]
3	.90	2 or 4 or 10	9	24	.8889	.9464	[.2500,.2500,.2500,.2500]
	.70	2	23	269	.6899	.8486	[.3762,.3119,.2223,.0897]
		4	21	225	.6985	.8529	[.4729,.2636,.1814,.0823]
		10	28	231	.6990	.8531	[.5089,.2456,.2229,.0228]
		15	23	462	.6996	.8534	[.2501,.3750,.2438,.1313]
	.65	2	40	715	.6430	.8249	[.4901,.2550,.1847,.0704]
		4	40	934	.6477	.8273	[.4789,.2606,.2059,.0548]
		10	38	670	.6487	.8278	[.4597,.2702,.1877,.0826]
		15	24	961	.6499	.8284	[.2795,.3603,.2443,.1161]
	.63	10	60	1823	.6287	.8176	[.4004,.2999,.2345,.0654]
		15	27	1734	.6300	.8183	[.2795,.3603,.2443,.1161]

Table 7: Testing the function `IsSubExp_opt` on the first Grigorchuk group. Initial weights are uniform.

7.2 Results on new groups

7.2.1 The group of Fig. 7 acting on a 6-letters alphabet

The function `IsSubExp_opt` starting from uniform weights with $\eta_{\text{tar}} = .67$ and update 10 yield the vector $[.3352, .1899, .1899, .2849]$. With these weights, the function `IsSubExp_fast` attained target $\eta = 0.645$ with radius 76 and an egg of size 69978. So this group has growth exponent at most $\alpha = .8034$. In view of the difficulty to find optimal weights, it is likely that the actual exponent is substantially smaller. However it would be a surprise if it were smaller than that of the Grigorchuk group.

7.2.2 The group of Fig. 8 acting on an 8-letters alphabet

Optimizations stronly suggest to use the following weights $[1., 0., 0.]$, i.e. to take only the involutive generator into account. Then the function `IsSubExp_fast` attained target $\eta = 0.819$ with radius 464 and an egg of size 2098. So this group has growth exponent at most $\alpha = .9124$. We believe this bound is close to optimal.

Note that using uniform weights, `IsSubExp_fast` attained target $\eta = 0.8299$, hence $\alpha = .9178$, with radius 157 and an egg of size 5690.

7.2.3 The group of Fig. 9 acting on a 7-letters alphabet

The function `IsSubExp_opt` with $\eta_{\text{tar}} = .9$ and update 10 returned the weights $[.3115, .2731, .4154]$. With this and target $.83$, the function `IsSubExp_ovi` yield the weights $[.3238, .2794, .3968]$ and $\eta = .8297$. Then `IsSubExp_fast` attained target $\eta = .8200$ with radius 91 and an egg of size 54727. So this group has growth exponent at most $\alpha = .9075$.

7.2.4 The group with X-shape Schreier graph, 9 states and 17 letters

This group was defined in Section 2.4. The function `IsSubExp_fast_uni` attained target $\eta = 0.9286$ with radius 18 and an egg of size 240039. So this group has growth exponent at most $\alpha = .9746$. It is likely that the actual growth exponent is much smaller, but it is already challenging to prove intermediate growth. The amount of data is so big that the functions `IsSubExp_ovi` and `IsSubExp_opt` cannot get to optimization on an ordinary laptop. This is why only uniform weights are used.

8 Super-polynomial growth

In this section we provide a folklore sufficient criterion that ensures super-polynomial growth of some automata groups. In particular, all the assumptions below are satisfied by the examples presented in Section 2.4.

Let G be an automaton group.

Assumption 8.1. There exists $A_1, \ldots, A_k$ finite subgroups of G with pairwise trivial intersection such that

- either $k \geq 3$ or $k = 2$ and A_1, A_2 are not both groups of size 2,

- the group G is generated by an automaton $\mathcal{A}$ with stateset $Q = A_1 \cup \cdots \cup A_k$.

Under Assumption 8.1, we take $S = Q \setminus \{e\}$. We denote the self-similar images of the generators as:

$$\forall s \in S, \quad \psi(s) = \left\langle s_{|x_1}, \ldots, s_{|x_d} \right\rangle \pi(s). \tag{7}$$

Assumption 8.2. For any word w in S^*, and for any section $x \in \Sigma$, we have

$$|w_{|x}|_S \leq \frac{|w|_S + 1}{2}.$$

Given our automaton $\mathcal{A}$, we construct an approximating sequence $(G^k)_{k \geq 0}$ as follows (we use superscripts to avoid confusion with sections which are denoted with subscripts).

Set $G^0 = A_1 * \cdots * A_k$ to be the free product of the finite subgroups of Assumption 8.1 and $S^0 = S$.

For each $k \geq 1$, the group G^k will be a subgroup of $G^{k-1} \wr_\Sigma \mathrm{Sym}(\Sigma)$ generated by a set S^k in canonical bijection with S. More precisely

$$\psi^k : G^k = \langle S^k \rangle \hookrightarrow G^{k-1} \wr_\Sigma \mathrm{Sym}(\Sigma)$$

where the generators are given by

$$\forall s^k \in S^k, \quad \psi^k(s^k) = \left\langle s^{k-1}_{|x_1}, \ldots, s^{k-1}_{|x_d} \right\rangle \pi(s) \quad \text{modelled on (7)}.$$

(Here, S^k, S^{k-1} and S are all in canonical bijection as generating sets of their appropriate groups.)

Lemma 8.3. *Under Assumptions 8.1 and 8.2, for any word w in S^* of length $|w|_S \leq 2^k$, the evaluation $w(S)$ in G is trivial if and only if the evaluation $w(S^k)$ in G^k is trivial.*

Proof. To check whether the evaluations are trivial, it is sufficient to check if their images in the iterated wreath products

$$G^k \overset{\psi^k \circ \cdots \circ \psi^1}{\hookrightarrow} G^0 \wr_{\Sigma^k} (\mathrm{Sym}(\Sigma) \wr_\Sigma \cdots \wr_\Sigma \mathrm{Sym}(\Sigma)) \text{ and } G \overset{\psi^{\circ k}}{\hookrightarrow} G \wr_{\Sigma^k} (\mathrm{Sym}(\Sigma) \wr_\Sigma \cdots \wr_\Sigma \mathrm{Sym}(\Sigma))$$

are trivial. But by construction of G^k, the images in $\mathrm{Sym}(\Sigma) \wr_\Sigma \cdots \wr_\Sigma \mathrm{Sym}(\Sigma)$ are the same. On the other hand for any section $\mathbf{u} \in \Sigma^k$, one has $w(S^k)_{|\mathbf{u}} = w(S)_{|\mathbf{u}}$ as words in S^0 and S respectively. Now Assumption 8.2 guarantees that if $|w|_S \leq 2^k$, then $|w_{|\mathbf{u}}|_S \leq 1$. By Assumption 8.1 and definition of G^0, the evaluations of $w(S^k)_{|\mathbf{u}}$ and $w(S)_{|\mathbf{u}}$ are either both trivial or both non-trivial. $\square$

Assumption 8.4. The self-similarity map $\psi : G \hookrightarrow G \wr_\Sigma \mathrm{Sym}(\Sigma)$ is surjective on the first section, i.e. for all g_1 in G, there exists g in G such that $g_{|x_1} = g_1$.

Proposition 8.5. *Assume G satisfies Assumptions 8.1, 8.2, 8.4 and $(\star)$ of Proposition 5.2, then G has intermediate growth.*

The following proof uses the notion of amenability of a group, which we will not define here. The reader can refer to [7] for instance. For our purpose, it is sufficient to know that subexponential growth implies amenability, that amenability passes to quotients and subgroups and that virtually free groups (e.g. the group G^0 under Assumption 8.1) are not amenable.

Proof. Property $(\star)$ of Proposition 5.2 guarantees subexponential growth. There remains to check super-polynomial growth. By Gromov's theorem, this amounts to prove that G is not virtually nilpotent. As finitely generated nilpotent groups are finitely presented, it is sufficient to prove that G is not finitely presented. Assume the contrary and take a finite presentation of G with respect to the generating set S. There exists some R such that all the defining relations have length $\leq R$. By Lemma 8.3 all these relations also hold in (G^k, S^k) for k large enough. Therefore G^k is a quotient of G hence amenable too. However by construction and Assumption 8.4, the group G^k contains a non-trivial free product $G^0 = A_1 * \cdots * A_k$ which is non amenable. This is a contradiction. $\square$

References

[1] A. Akhavi, I. Klimann, S. Lombardy, J. Mairesse, and M. Picantin. On the finiteness problem for automaton (semi)groups. *International Journal of Algebra and Computation*, 22(6):1–26, 2012.

[2] V. Aleshin, S. Finite automata and the Burnside problem for periodic groups. *Mat. Zametki*, 11:319–328, 1972.

[3] L. Bartholdi. The growth of Grigorchuk's torsion group. *Internat. Math. Res. Notices*, 20:1049–1054, 1998.

[4] L. Bartholdi. A Wilson group of non-uniformly exponential growth. *C. R. Math. Acad. Sci. Paris*, 336(7):549–554, 2003.

[5] L. Bartholdi. *FR – GAP package "Computations with functionally recursive groups", Version 2.2.1*, 2015. *http://www.gap-system.org/Packages/fr.html*.

[6] L. Bartholdi. Algorithmic decidability of Engel's property for automaton groups. In *Computer Science - Theory and Applications - 11th International Computer Science Symposium in Russia, CSR 2016, St. Petersburg, Russia, June 9-13, 2016, Proceedings*, pages 29–40, 2016.

[7] L. Bartholdi. *Amenability of groups and G-sets*. Sequences, Groups, and Number Theory. Springer International Publishing, 2018.

[8] L. Bartholdi and A. Erschler. Growth of permutational extensions. *Invent. Math.*, 189(2):431–455, 2012.

[9] L. Bartholdi and A. Erschler. Groups of given intermediate word growth. *Annales de l'institut Fourier*, 64(5):2003–2036, 2014.

[10] L. Bartholdi, T. Godin, I. Klimann, and M. Picantin. A new hierarchy for automaton semigroups. In *Proc. 23rd CIAA*, LNCS, 2018.

[11] L. Bartholdi and Z. Šunić. On the word and period growth of some groups of tree automorphisms. *Communications in Algebra*, 29-11(11):4923–4964, 2001.

[12] S. Blackburn, C. Cid, and C. Mullan. *Group Theory in Cryptography*, volume 387 of *LMS Lecture Note Series*, pages 133–149. The London Mathematical Society, 2011.

[13] I. V. Bondarenko, N. V. Bondarenko, S. N. Sidki, and F. R. Zapata. On the conjugacy problem for finite-state automorphisms of regular rooted trees. *Groups Geom. Dyn.*, 7(2):323–355, 2013. With an appendix by R. M. Jungers.

[14] J. Brieussel. Amenability and non-uniform growth of some directed automorphism groups of a rooted tree. *Math. Z.*, 263:265–293, 2009.

[15] J. Brieussel. Growth behaviors in the range $e^{r^{\alpha}}$. *Afrika Matematika*, 25(4):1143–1163, 2014.

[16] J. Brieussel. An automata group of intermediate growth and exponential activity. *Journal of Group Theory*, 21(4):573–578, 2018.

[17] J. Conway, R. Curtis, S. Norton, R. Parke, and R. Wilson. *Atlas of Finite Groups: Maximal Subgroups and Ordinary Characters for Simple Groups*. Clarendon Press, 1985.

[18] F. Cotton. *Chemical Applications of Group Theory.* A Wiley-Interscience publication. Wiley India, 2003.

[19] P. de la Harpe. *Topics in Geometric Group Theory.* Chicago Lectures in Mathematics. University of Chicago Press, 2000.

[20] A. Egri-Nagy, A. R. Francis, and V. Gebhardt. Bacterial genomics and computational group theory: The BioGAP package for GAP. In H. Hong and C. Yap, editors, *2014*, pages 67–74, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[21] A. Erschler. Boundary behavior for groups of subexponential growth. *Ann. Math.*, 160:1183–1210, 2004.

[22] A. Erschler and T. Zheng. Growth of periodic Grigorchuk groups. *ArXiv e-prints*, Feb. 2018.

[23] J. Fabrikowsky and N. Gupta. On groups with sub-exponential growth functions, ii. *J. Indian Math. Soc.(NS)*, 56:217–228, 1991.

[24] The GAP Group. `GAP` – *Groups, Algorithms, and Programming*, 2015. http://www.gap-system.org.

[25] E. S. Golod. On nil-algebras and finitely residual groups. *Izv. Akad. Nauk SSSR. Ser. Mat.*, 28:273–276, 1964.

[26] E. S. Golod and I. Shafarevich. On the class field tower. *Izv. Akad. Nauk SSSR Ser. Mat.*, 28:261–272, 1964.

[27] R. I. Grigorchuk. Milnor's problem on the growth of groups. *Sov. Math. Dokl.*, 28:23–26, 1983.

[28] R. I. Grigorchuk. Degrees of growth of finitely generated groups, and the theory of invariant means. *Izvestiya: Mathematics*, 25(2):259–300, 1985.

[29] R. I. Grigorchuk. On the growth degrees of p-groups and torsion-free groups. *Math. USSR Sbornik*, 54(1):185–205, 1986.

[30] R. I. Grigorchuk. *Milnor's problem on the growth of groups and its consequences*, pages 705–774. Princeton University Press, 2014.

[31] R. I. Grigorchuk and I. Pak. Groups of intermediate growth: an introduction. In *Enseign. Math.* Citeseer, 2008.

[32] M. Gromov. Groups of polynomial growth and expanding maps. *Publ. Math., Inst. Hautes Étud. Sci*, pages 53–73, 1981.

[33] H. Helfgott. Isomorphismes de graphes en temps quasi-polynomial (d'après Babai et Luks, Weisfeiler-Leman,...). *Astérisque*, (1125), 2018.

[34] I. Klimann. To Infinity and Beyond. In I. Chatzigiannakis, C. Kaklamanis, D. Marx, and D. Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 131:1–131:12, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[35] I. Klimann, J. Mairesse, and M. Picantin. Implementing computations in automaton (semi)groups. In *Proc. 17th CIAA*, volume 7381 of *LNCS*, pages 240–252, 2012.

[36] A. Mann. *How Groups Grow*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2011.

[37] J. Milnor. Growth of finitely generated solvable groups. *J. Diff. Geom.*, 2:447–449, 1968.

[38] J. Milnor. A note on curvature and fundamental group. *J. Diff. Geom.*, 2:1–7, 1968.

[39] J. Milnor. Problem 5603. *Amer. Math. Monthly*, 75(6):685–686, 1968.

[40] B. Mohammadi. Optimal transport, shape optimization and global minimization. *Comptes Rendus Mathematique*, 344(9):591 – 596, 2007.

[41] B. Mohammadi and P. Redont. Improving the identification of general pareto fronts by global optimization. *Comptes Rendus Mathematique*, 347(5):327 – 331, 2009.

[42] R. Muchnik and I. Pak. On growth of Grigorchuk groups. *International Journal of Algebra and Computation*, 11(01):1–17, 2001.

[43] Y. Muntyan and D. Savchuk. *AutomGrp – GAP package for computations in self-similar groups and semigroups, Version 1.2.4*, 2014. *http://www.gap-system.org/Packages/automgrp.html*.

[44] V. Nekrashevych. *Self-similar groups*, volume 117 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2005.

[45] V. Nekrashevych. Palindromic subshifts and simple periodic groups of intermediate growth. *Ann. Math. (2)*, 187(3):667–719, 2018.

[46] P. S. Novikov and S. I. Adjan. Infinite periodic groups I. *Izv. Akad. Nauk SSSR Ser. Mat.*, 32:212–244, 1968.

[47] F. Olukoya. The growth rates of automaton groups generated by reset automata. *ArXiv e-prints*, Aug. 2017.

[48] I. I. Reznikov. On 2-state mealy automata of polynomial growth. *Algebra and Discrete Mathematics*, 4, 2003.

[49] I. I. Reznikov and V. I. Sushchanskii. A software system for growth analysis of Mealy automata. *Cybernetics and Systems Analysis*, 42(2):265–276, Mar 2006.

[50] S. Sidki. Automorphisms of one-rooted trees: growth, circuit structure, and acyclicity. *Journal of Mathematical Sciences (New York)*, 100(1):1925–1943, 2000. Algebra, 12.

[51] R. Slansky. Group theory for unified model building. *Physics Reports*, 79(1):1 – 128, 1981.

[52] A. Svarc. A volume invariant of coverings. *Dokl. Akad. Nauk. SSSR*, 105:32–34, 1955.

[53] J. Tits. Free subgroups in linear groups. *J. Algebra*, 20:250–270, 1972.

[54] Y. Vorobets. Notes on the Schreier graphs of the Grigorchuk group. In R. Amer. Math. Soc., Providence, editor, *Dynamical systems and group actions*, volume 567, pages 221–248, 2012.

[55] J. S. Wilson. Further groups that do not have uniformly exponential growth. *J. Algebra*, 279:292–301, 2004.

[56] J. S. Wilson. On exponential growth and uniformly exponential growth for groups. *Invent. Math.*, 155(2):287–303, 2004.

[57] J. A. Wolf. Growth of finitely generated solvable groups and curvature of Riemannian manifolds. *J. Diff. Geom.*, 2:421–446, 1968.

[58] A. Zuk. Groupes engendrés par des automates. Seminaire Bourbaki 971, 2006.

Jérémie Brieussel — IMAG, Université de Montpellier — jeremie.brieussel@umontpellier.fr

Thibault Godin — IMAG, Université de Montpellier — thib.godin@gmail.com

Bijan Mohammadi — IMAG, Université de Montpellier — bijan.mohammadi@umontpellier.fr

A Figures of automata groups and their Schreier graphs
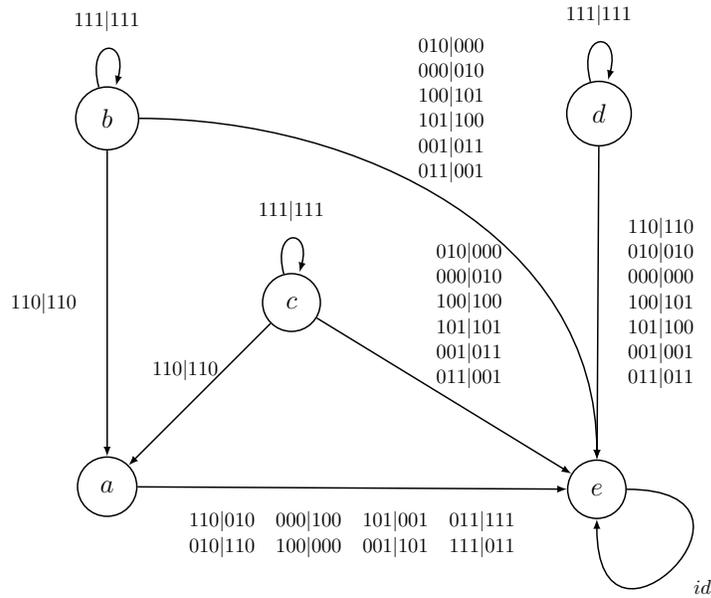
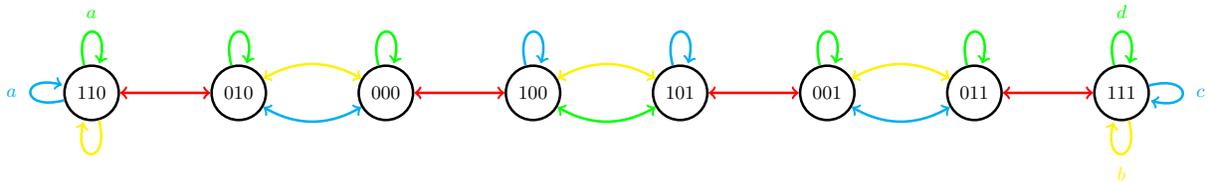Figure 5: The automata generating the first Grigorchuk group on level 3.

Figure 6: The Schreier graph on level 3 of the automata generating the first Grigorchuk group.
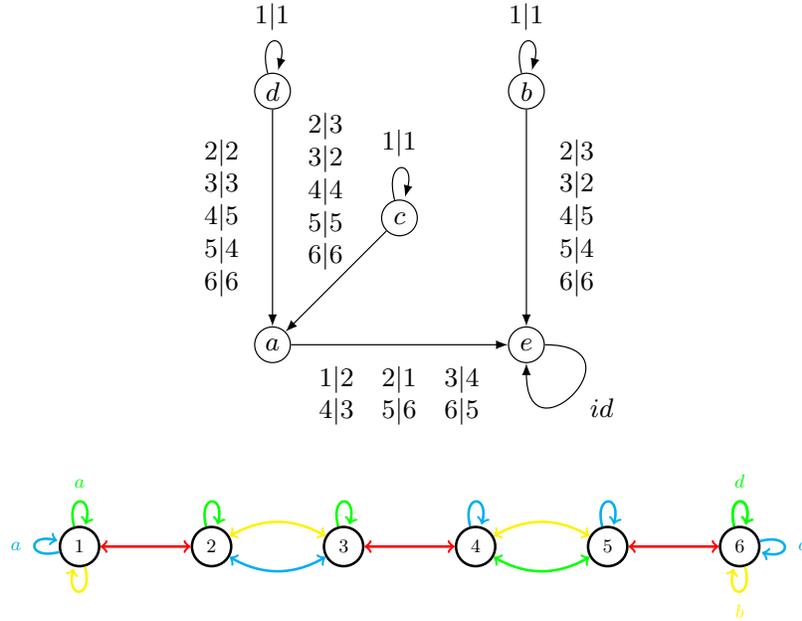
Figure 7: A novel automata generating a group of intermediate growth (on the top) and its Schreier graph on level 1 (on the bottom). Notice the similarities with the Grigorchuk group on Fig. 2.
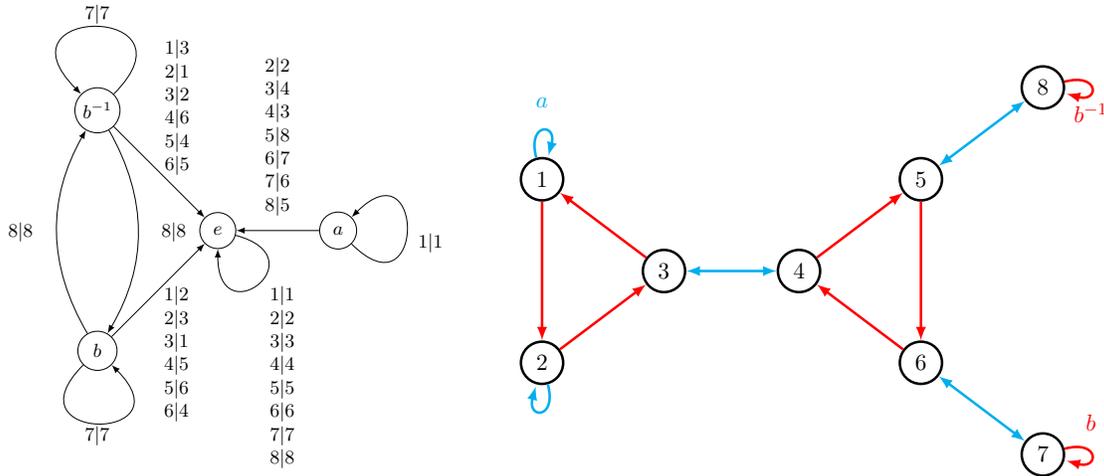

Figure 8: The automaton $a = \langle a, e, e, e, e, e, e, e \rangle (3,4)(5,8)(6,7); b = \langle e, e, e, e, e, e, b, b^{-1} \rangle (1,2,3)(4,5,6)$ (on the left) from [16] and its Schreier graph on level 1 (on the right).
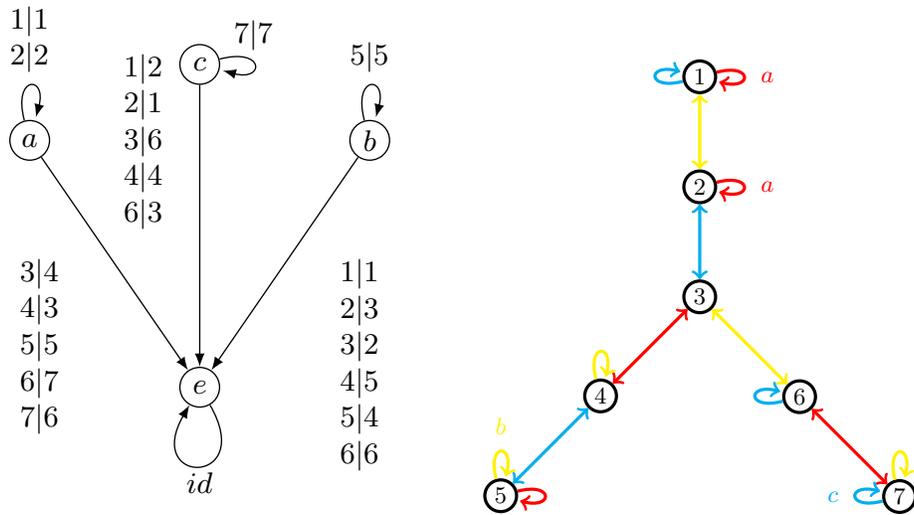
Figure 9: A novel automata generating a group of intermediate growth (on the left) and its Schreier graph on level 1 (on the right).