Streaming Word Problems

Markus Lohrey
Universität Siegen, Germany

Lukas Lück
Universität Siegen, Germany

Abstract

We study deterministic and randomized streaming algorithms for word problems of finitely generated groups. For finitely generated linear groups, metabelian groups and free solvable groups we show the existence of randomized streaming algorithms with logarithmic space complexity for their word problems. We also show that the class of finitely generated groups with a logspace randomized streaming algorithm for the word problem is closed under several group theoretical constructions: finite extensions, direct products, free products and wreath products by free abelian groups. We contrast these results with several lower bound. An example of a finitely presented group, where the word problem has only a linear space randomized streaming algorithm, is Thompson's group F.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases word problems for groups, streaming algorithms

Digital Object Identifier 10.4230/LIPIcs.MFCS.2022.72

Related Version *Full Version*: https://arxiv.org/abs/2202.04060

Funding *Markus Lohrey*: Funded by DFG project LO 748/12-1.

1 Introduction

The word problem for a finitely generated group G is the following computational problem: Fix a finite set of generators Σ for G (which means that every element of G can be written as a finite product of elements from Σ. The input for the word problem is a finite word $a_1a_2\cdots a_n$ over the alphabet Σ and the question is whether this word evaluates to the group identity of G. The word problem was introduced by Dehn in 1911 [10]. It is arguably the most important computational problem in group theory and has been studied by group theorists as well as computer scientists; see [29] for a good survey. In recent years, complexity theoretic investigations of word problems moved into the focus. For many important classes of groups it turned out that the word problem belongs to low-level complexity classes. The first result in this direction was proved by Lipton and Zalcstein [23] (if the field F has characteristic zero) and Simon [36] (if the field F has prime characteristic): if G is a finitely generated linear group over an arbitrary field F (i.e., a finitely generated group of invertible matrices over F), then the word problem for G can be solved in deterministic logarithmic space. Related results can be found in [20, 41].

The word problem of a group G with a finite generating set Σ can be identified with a formal language $\mathsf{WP}(G,\Sigma)$ consisting of all words over the alphabet Σ that evaluate to the group identity of G. Language theoretic aspects of the word problem have been studied intensively in the past. For instance, Anissimov and Seifert [2] showed that $\mathsf{WP}(G,\Sigma)$ is regular if and only if G is finite, and Muller and Schupp showed that $\mathsf{WP}(G,\Sigma)$ is context-free [31] if and only if G is virtually free,[1] see [18] for an overview.

[1] If C is a property or class of groups, then a group is virtually C if it is a finite extension of a C-group.

In this paper we initiate the study of streaming algorithms for word problems. These are algorithms that do not have random access on the whole input. Instead, the k-th input symbol is only available at time k [1]. Typically, streaming algorithms are randomized and have a bounded error probability. Usually, one is interested in the space used by a streaming symbol, but also update times have been investigated. Clearly, every regular language has a streaming algorithm with constant space. Randomized streaming algorithms for context-free languages have been studied in [4, 7, 12, 26].

Let us now explain the main results of this paper. For a finitely generated group G with generating set Σ, the *deterministic (resp., randomized) streaming space complexity* of $\mathsf{WP}(G, \Sigma)$ is the space complexity of the best deterministic (resp., randomized) streaming algorithm for $\mathsf{WP}(G, \Sigma)$. The concrete choice of the generating set has only a minor influence on the deterministic (resp., randomized) streaming space complexity of $\mathsf{WP}(G, \Sigma)$; see Lemma 2 for a precise statement. In statements where the influence of the generating set on the streaming space complexity is blurred by the Landau notation, we speak of the deterministic/randomized streaming space complexity of the word problem of G or simply the deterministic/randomized streaming space complexity of G.

The deterministic streaming space complexity of $\mathsf{WP}(G, \Sigma)$ is directly linked to the growth function $\gamma_{G,\Sigma}(n)$ of the group G. The latter is the number of different group elements of G that can be represented by words over the generating set Σ of length at most n (also here the generating set Σ only has a minor influence). The deterministic streaming space complexity of the word problem for G turns out to be $\log_2 \gamma_{G,\Sigma}(n/2)$ up to a small additive constant (Theorem 3). The growth of finitely generated groups is a well investigated topic in geometric group theory. A famous theorem of Gromov says that a finitely generated group has polynomial growth if and only if it is virtually nilpotent; see [9, 28] for a discussion. Theorem 3 reduces all questions about the deterministic streaming space complexity of word problems to questions about growth functions. Due to this, we mainly study randomized streaming algorithms for word problems in this paper.

In the randomized setting, the growth of G still yields a lower bound: The randomized streaming space complexity of the word problem of G is lower bounded by $\Omega(\log\log \gamma_{G,\Sigma}(n/2))$ (Theorem 4). A large class of groups, where this lower bound can be exactly matched by an upper bound are finitely generated linear groups. Recall that Lipton and Zalcstein [23] and Simon [36] showed that the word problem of a finitely generated linear group can be solved in logarithmic space. Their algorithm can be turned into a randomized streaming algorithm with logarithmic space complexity. In order to plug these streaming algorithms into closure results for randomized streaming space complexity (that are discussed below) we need an additional property that we call ϵ-injectivity. Roughly speaking, a randomized streaming algorithm for a finitely generated group G with generating set Σ is *ϵ-injective* if for all words $u, v \in \Sigma^*$ of length at most n we have that: (i) if u and v evaluate to the same element of G then with probability at least $1 - \epsilon$, u and v lead to the same memory state of the streaming algorithm, and (ii) if u and v evaluate to different elements of G then with probability at least $1 - \epsilon$, u and v lead to different memory states of the streaming algorithm; see Section 5. We then show that for every finitely generated linear group G there is a randomized ϵ-injective streaming algorithm with space complexity $\mathcal{O}(\log n)$ (Theorem 8). If G is moreover virtually nilpotent, then the space complexity can be further reduced to $\mathcal{O}(\log\log n)$. In fact, using a known gap theorem for the growth of linear groups [30, 42], it turns out that the randomized streaming space complexity of the word problem for a finitely generated linear group G is either $\Theta(\log\log n)$ (if G is virtually nilpotent) or $\Theta(\log n)$ (if G is not virtually nilpotent), see Theorem 11.

For non-linear groups the situation turns out to be more difficult. We show that the randomized streaming space complexity of word problems is preserved by certain group constructions including finite extensions (Theorem 10), direct products (Lemma 12), free products (Theorem 20) and wreath products by free abelian groups (Theorem 17). For the latter two constructions we also get an additional additive term $\Theta(\log n)$ in the space bounds. For a wreath product $A \wr G$ with A free abelian (resp., a free product $G * H$) it is also important that we start with an ϵ-injective randomized streaming algorithm for G (and H). Using these transfer results we obtain also non-linear groups with a logarithmic randomized streaming space complexity, e.g., metabelian groups (Corollary 13) and free solvable groups (Corollary 18).

In the last section of the paper, we prove lower bounds for the randomized streaming space complexity of word problems. For wreath products of the form $G \wr S$ such that G is non-abelian and S is infinite, we can show that the randomized streaming space complexity is $\Theta(n)$ by a reduction from the randomized communication complexity of disjointness (Theorem 21). A concrete finitely presented group with randomized streaming space complexity $\Theta(n)$ is Thompson's group F (Corollary 22). Thompson's groups F (introduced by Richard Thompson in 1965) belongs due to its unusual properties to the most intensively studied infinite groups; see e.g. [8]. From a computational perspective it is interesting to note that F is co-context-free (i.e., the set of all non-trivial words over any set of generators is a context-free language) [22]. This implies that the word problem for Thompson's group is in LogCFL. To the best of our knowledge no better upper complexity bound is known. Finally, we consider the famous Grigorchuk group G [14], which was the first example of a group with intermediate word growth as well as the first example of a group that is amenable but not elementary amenable. We show that the deterministic streaming space complexity of G is $\mathcal{O}(n^{0.768})$, whereas the randomized streaming space complexity of G is $\Omega(n^{0.5})$ (Theorem 23).

Related results. In this paper, we are only interested in streaming algorithms for a fixed infinite group. Implicitly, streaming algorithms for finite groups are studied in [13]. Obviously, every finite group has a deterministic streaming space complexity $\mathcal{O}(\log |G|)$.[2] In [13], it is shown that for the group $G = \mathsf{SL}(2, \mathbb{F}_p)$ this upper bound is matched by a lower bound, which even holds for the randomized streaming space complexity. More precisely, Gowers and Viola study the communication cost of the following problem: Alice receives a sequence of elements $a_1, \ldots, a_n \in G$, Bob receives a sequence of elements $b_1, \ldots, b_n \in G$ and they are promised that the interleaved product $a_1 b_1 \cdots a_n b_n$ is either 1 or some fixed element $g \in G \setminus \{1\}$ and their job is to determine which of these two cases holds. For $G = \mathsf{SL}(2, \mathbb{F}_p)$ it is shown that the randomized communication complexity of this problem is $\Theta(\log |G| \cdot n)$ (the upper bound is trivial). From this it follows easily that the randomized streaming space complexity of $\mathsf{SL}(2, \mathbb{F}_p)$ is $\Omega(\log |G|)$.

2 Streaming algorithms

For integers $a < b$ let $[a, b] = \{a, a+1, \ldots, b\}$. Fix a finite alphabet Σ. For a word $w \in \Sigma^*$ let $|w|$ be its length and let $\Sigma^{\leq n} = \{w \in \Sigma^* : |w| \leq n\}$ be the set of words of length at most n.

In the following we introduce probabilistic finite automata [33, 34] as a model for randomized streaming algorithms. A *probabilistic finite automaton* (PFA) $\mathcal{A} = (Q, \Sigma, \iota, \rho, F)$ consists of a finite set of states Q, an alphabet Σ, an *initial state distribution* $\iota \colon Q \to \{r \in$

[2] In our setting, $|G|$ would be a constant, but for the moment let us make the dependence on the finite group G explicit.

$\mathbb{R}\colon 0 \leq r \leq 1\}$, a *transition probability function* $\rho\colon Q \times \Sigma \times Q \to \{r \in \mathbb{R}\colon 0 \leq r \leq 1\}$ and a set of final states $F \subseteq Q$ such that $\sum_{q \in Q} \iota(q) = 1$ and $\sum_{q \in Q} \rho(p, a, q) = 1$ for all $p \in Q$, $a \in \Sigma$. If ι and ρ map into $\{0, 1\}$, then $\mathcal{A}$ is a *deterministic finite automaton* (DFA). If only ρ is required to map into $\{0, 1\}$, then $\mathcal{A}$ is called *semi-probabilitistic*. A *run* on a word $a_1 \cdots a_m \in \Sigma^*$ in $\mathcal{A}$ is a sequence $\pi = (q_0, a_1, q_1, a_2, \ldots, a_m, q_m)$ where $q_0, \ldots, q_m \in Q$. Given such a run π in $\mathcal{A}$ we define $\rho_\iota(\pi) = \iota(q_0) \cdot \prod_{i=1}^{n} \rho(q_{i-1}, a_i, q_i)$. For each $w \in \Sigma^*$ the function ρ_ι is a probability distribution on the set $\mathrm{Runs}(w)$ of all runs of $\mathcal{A}$ on w. A run $\pi = (q_0, a_1, \ldots, a_m, q_m)$ is *correct* with respect to a language $L \subseteq \Sigma^*$ if $q_m \in F \Leftrightarrow a_1 \cdots a_m \in L$ holds. The *error probability* of $\mathcal{A}$ on w for L is

$$\epsilon(\mathcal{A}, w, L) = \sum \{\rho_\iota(\pi) : \pi \in \mathrm{Runs}(w) \text{ is not correct w.r.t. } L\}.$$

If $\mathcal{A}$ is semi-probabilitistic then we can identify ρ with a mapping $\rho : Q \times \Sigma \to Q$, where $\rho(p, a)$ is the unique state q with $\rho(p, a, q) = 1$. This mapping ρ is extended to a mapping $\rho : Q \times \Sigma^* \to Q$ in the usual way: $\rho(p, \varepsilon) = p$ and $\rho(p, aw) = \rho(\rho(p, a), w)$. We then obtain

$$\epsilon(\mathcal{A}, w, L) = 1 - \sum \{\iota(q) : q \in Q, \delta(q, w) \in F \Leftrightarrow w \in L\}.$$

A (non-uniform) *randomized streaming algorithm* is a sequence $\mathcal{R} = (\mathcal{A}_n)_{n \geq 0}$ of PFA $\mathcal{A}_n$ over the same alphabet Σ. If every $\mathcal{A}_n$ is deterministic (resp., semi-probabilitistic), we speak of a *deterministic (resp., semi-randomized)* streaming algorithm.

Let $0 \leq \epsilon \leq 1$ be an error probability. A randomized streaming algorithm $\mathcal{R} = (\mathcal{A}_n)_{n \geq 0}$ is *ϵ-correct* for a language $L \subseteq \Sigma^*$ if for every $n \geq 0$ and every word $w \in \Sigma^{\leq n}$ we have $\epsilon(\mathcal{A}_n, w, L) \leq \epsilon$. We say that $\mathcal{R}$ is a *randomized streaming algorithm for L* if it is $1/3$-correct for L. The choice of $1/3$ for the error probability is not important. Using a standard application of the Chernoff bound one can reduce the error probability from $1/3$ to every constant. If $\mathcal{R}$ is deterministic and 0-correct for L then we say that $\mathcal{R}$ is a *deterministic streaming algorithm for L*. The *space complexity* of the randomized streaming algorithm $\mathcal{R} = (\mathcal{A}_n)_{n \geq 0}$ is the function $s(\mathcal{R}, n) = \lceil \log_2 |Q_n| \rceil$, where Q_n is the state set of $\mathcal{A}_n$. The motivation for this definition is that states of Q_n can be encoded by bit strings of length at most $\lceil \log_2 |Q_n| \rceil$. The *deterministic/randomized streaming space complexity of the language L* is the smallest possible function $s(\mathcal{R}, n)$, where $\mathcal{R}$ is a deterministic/randomized streaming algorithm for L. By a result of Rabin [34, Theorem 3], the deterministic streaming space complexity of a language L is bounded by $2^{\mathcal{O}(S(n))}$, where $S(n)$ is the randomized streaming space complexity of L.

The deterministic streaming space complexity of a language L is directly linked to the *automaticity of L*. The automaticity of $L \subseteq \Sigma^*$ is the the function $A_L(n)$ that maps n to the number of states of a smallest DFA $\mathcal{A}_n$ such that for all words $w \in \Sigma^{\leq n}$ we have: $w \in L$ if and only if w is accepted by $\mathcal{A}_n$. Hence, the deterministic streaming space complexity of L is exactly $\lceil \log_2 A_L(n) \rceil$. The automaticity of languages was studied in [35]. Interesting in our context is the following result of Karp [19]: if L is a non-regular language then $A_L(n) \geq (n + 3)/2$ for infinitely many n. Hence, for every non-regular language the deterministic streaming space complexity of L is at least $\log_2(n) - c$ for a constant c and infinitely many n. Another related measure is the online space complexity from [11], which is defined by the growth function of an infinite automaton for a language L.

Note that our concept of streaming algorithms is non-uniform in the sense that for every input length n we have a separate streaming algorithm $\mathcal{A}_n$. This makes lower bounds stronger. On the other hand, the streaming algorithms that we construct for concrete groups will be uniform in the sense that the streaming algorithms $\mathcal{A}_n$ follow a common pattern. The following result uses non-uniformity in a crucial way; its proof (see [25, Theorem 3.1]) is similar to Newman's theorem on public versus private coins in communication complexity, see e.g. [21].

▶ **Theorem 1.** *Let $\mathcal{R}$ be a randomized streaming algorithm such that $s(\mathcal{R}, n) \geq \Omega(\log n)$ and $\mathcal{R}$ is ϵ-correct for a language L. Then there exists a semi-randomized streaming algorithm $\mathcal{S}$ such that $s(\mathcal{S}, n) = \Theta(s(\mathcal{R}, n))$ and $\mathcal{S}$ is 2ϵ-correct for the language L.*

3 Groups and word problems

For a group G and a subset $\Sigma \subseteq G$, we denote with $\langle \Sigma \rangle$ the subgroup of G generated by Σ. It is the set of all products of elements from $\Sigma \cup \Sigma^{-1}$. We only consider *finitely generated (f.g.) groups* G, for which there is a finite set $\Sigma \subseteq G$ such that $G = \langle \Sigma \rangle$; such a set Σ is called a *finite generating set* for G. If $\Sigma = \Sigma^{-1}$ then we say that Σ is a *finite symmetric generating set* for G. In the following we assume that all finite generating sets are symmetric. Every word $w \in \Sigma^*$ evaluates to a group element $\pi_G(w)$ in the natural way; here $\pi_G : \Sigma^* \to G$ is the canonical morphism from the free monoid Σ^* to G. Instead of $\pi_G(u) = \pi_G(v)$ we also write $u \equiv_G v$.

Let $C(G, \Sigma)$ be the Cayley graph of G with respect to the finite symmetric generating set Σ. It is the edge-labelled graph whose vertex set is G and that has an a-labelled edge from $\pi_G(u)$ to $\pi_G(ua)$ for all $u \in \Sigma^*$ and $a \in \Sigma$. Let $\mathsf{WP}(G, \Sigma) = \{w \in \Sigma^* \mid \pi_G(w) = 1\}$ be the *word problem for G* with respect to the generating set Σ.

We are interested in streaming algorithms for words problems $\mathsf{WP}(G, \Sigma)$. The following lemma is easy to prove; see [25, Lemma 4.1].

▶ **Lemma 2.** *Let Σ_1 and Σ_2 be finite symmetric generating sets for the group G and let $s_i(n)$ be the deterministic/randomized streaming space complexity of $\mathsf{WP}(G, \Sigma_i)$. Then there exists a constant c that depends on G, Σ_1 and Σ_2 such that $s_1(n) \leq s_2(c \cdot n)$.*

By Lemma 2, the dependence of the streaming space complexity from the generating set is often blurred by the use of the $\mathcal{O}$-notation. In such situations we will speak of the deterministic/randomized streaming space complexity for the group G (instead of the deterministic/randomized streaming space complexity of the language $\mathsf{WP}(G, \Sigma)$).

4 Streaming algorithms for word problems and growth

Let G be a finitely generated group and let Σ be a finite symmetric generating set for G. For $n \in \mathbb{N}$ let $B_{G,\Sigma}(n) = \pi_G(\Sigma^{\leq n}) \subseteq G$ be the ball of radius n in the Cayley-graph of G with center 1. The growth function $\gamma_{G,\Sigma} : \mathbb{N} \to \mathbb{N}$ is the function with $\gamma_{G,\Sigma}(n) = |B_{G,\Sigma}(n)|$. The deterministic streaming space complexity of G is completely determined by the growth of G:

▶ **Theorem 3.** *Let G be a finitely generated infinite group and let Σ be a finite symmetric generating set for G. Define the function $S(n)$ by $S(n) = \gamma_{G,\Sigma}(\lfloor n/2 \rfloor) + (n \bmod 2)$. Then, the deterministic streaming space complexity of $\mathsf{WP}(G, \Sigma)$ is $\lceil \log_2 S(n) \rceil$.*

Proof. We start with the upper bound in case n is even. In the following we identify the ball $B_{G,\Sigma}(n/2)$ with its induced subgraph of the Cayley graph $C(G, \Sigma)$. We define a deterministic finite automaton $\mathcal{A}_n$ by taking the edge-labelled graph $B_{G,\Sigma}(n/2)$ with the initial and unique final state 1. It can be viewed as a partial DFA in the sense that for every $g \in B_{G,\Sigma}(n/2)$ and every $a \in \Sigma$, g has at most one outgoing edge labelled with a (that leads to $g \cdot a$ if $g \cdot a \in B_{G,\Sigma}(n/2)$). In order to add the missing transitions we choose an element $g_f \in B_{G,\Sigma}(n/2) \setminus B_{G,\Sigma}(n/2 - 1)$ (here, we set $B_{G,\Sigma}(-1) = \emptyset$). Such an element exists because G is infinite. If $g \in B_{G,\Sigma}(n/2)$ has not outgoing a-labelled edge in $B_{G,\Sigma}(n/2)$ then we add an a-labelled edge from g to g_f. We call those edges *spurious*. The resulting DFA is $\mathcal{A}_n$.

We claim that for every word $w \in \Sigma^{\leq n}$, w is accepted by $\mathcal{A}_n$ if and only if $w \in \mathsf{WP}(G, \Sigma)$. This is clear, if no spurious edge is traversed while reading w into $\mathcal{A}_n$. In this case, after reading w, we end up in state $\pi_G(w)$. Now assume that a spurious edge is traversed while reading w into $\mathcal{A}_n$ and let x be the shortest prefix of w such that a spurious edge is traversed while reading the last symbol of x. Let us write $w = xy$. We must have $|x| > n/2$ and $\pi_G(x) \notin B_{G,\Sigma}(n/2)$. Moreover, $|y| < n - n/2 = n/2$. Since $\pi_G(x) \notin B_{G,\Sigma}(n/2)$, we have $w = xy \notin \mathsf{WP}(G, \Sigma)$. Moreover, w is rejected by $\mathcal{A}_n$, because x leads in $\mathcal{A}_n$ from the initial state 1 to state g_f and there is no path of length at most $n/2 - 1$ from g_f back to the final state 1.

For the case that n is odd, we take the ball $B_{G,\Sigma}(\lfloor n/2 \rfloor)$. Instead of adding spurious edges we add a failure state f. If $g \in B_{G,\Sigma}(\lfloor n/2 \rfloor)$ has no outgoing a-labelled edge in $B_{G,\Sigma}(\lfloor n/2 \rfloor)$, then we add an a-labelled edge from g to f. Moreover, for every $a \in \Sigma$ we add an a-labelled loop at state f. As for the case n even, one can show that the resulting DFA accepts a word $w \in \Sigma^{\leq n}$ if and only if $w \in \mathsf{WP}(G, \Sigma)$.

For the lower bound, let $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ be a smallest DFA such that for every word $w \in \Sigma^{\leq n}$, w is accepted by $\mathcal{A}$ if and only if $w \in \mathsf{WP}(G, \Sigma)$. We have to show that $|Q| \geq S(n)$. Let us consider two words $u, v \in \Sigma^*$ of length at most $\lfloor n/2 \rfloor$ such that $u \not\equiv_G v$ and $\delta(q_0, u) = \delta(q_0, v)$. We then have $uv^{-1} \notin \mathsf{WP}(G, \Sigma)$ and $vv^{-1} \in \mathsf{WP}(G, \Sigma)$. On the other hand, we have $\delta(q_0, uv^{-1}) = \delta(q_0, vv^{-1})$, which is a contradiction (note that $|uv^{-1}|, |vv^{-1}| \leq n$). Hence, if $\delta(q_0, u) = \delta(q_0, v)$ for two words $u, v \in \Sigma^*$ of length at most $\lfloor n/2 \rfloor$, then $u \equiv_G v$.

Let $Q' = \{\delta(q_0, w) \mid w \in \Sigma^*, |w| \leq \lfloor n/2 \rfloor\} \subseteq Q$. The previous paragraph shows that $|Q'| \geq \gamma_{G,\Sigma}(\lfloor n/2 \rfloor)$. If n is even then $\lfloor n/2 \rfloor = n/2$ and we are done. So, let us assume that n is odd.

If $|Q| > \gamma_{G,\Sigma}(\lfloor n/2 \rfloor)$ then we are again done. So, let us assume that $Q = Q'$ and $|Q| = \gamma_{G,\Sigma}(\lfloor n/2 \rfloor)$. Then, to every state $q \in Q$ we can assign a unique group element $g_q \in B_{G,\Sigma}(\lfloor n/2 \rfloor)$ such that for every word $w \in \Sigma^*$ with $|w| \leq \lfloor n/2 \rfloor$ we have $\delta(q_0, w) = q$ if and only if $\pi_G(w) = g_q$. The mapping $q \mapsto g_q$ is a bijection between Q and $B_{G,\Sigma}(\lfloor n/2 \rfloor)$.

Let us now take a state $q \in Q$ and a generator $a \in \Sigma$ such that $g_q \cdot a \notin B_{G,\Sigma}(\lfloor n/2 \rfloor)$. Such a state and generator must exist since G is infinite. Let $u, v \in \Sigma^*$ be words of length at most $\lfloor n/2 \rfloor$ such that $\delta(q_0, u) = q$ and $\delta(q_0, v) = \delta(q, a) = \delta(q_0, ua)$. We obtain $\delta(q_0, vv^{-1}) = \delta(q_0, uav^{-1})$. But $vv^{-1} \in \mathsf{WP}(G, \Sigma)$ and $uav^{-1} \notin \mathsf{WP}(G, \Sigma)$ since $\pi_G(uav^{-1}) = g_q \cdot a \cdot \pi_G(v^{-1})$ and $g_q \cdot a \notin B_{G,\Sigma}(\lfloor n/2 \rfloor)$, $\pi_G(v^{-1}) \in B_{G,\Sigma}(\lfloor n/2 \rfloor)$. This is a contradiction since vv^{-1} and uav^{-1} both have length at most n. ◀

The growth of f.g. groups is well-studied and Theorem 3 basically closes the chapter on deterministic streaming algorithms for word problems. Hence, in the rest of the paper we focus on randomized streaming algorithms. Here, we can still prove a lower bound (that will turn out to be sharp in some cases but not always) using the randomized one-way communication complexity of the equality problem; see [25, Theorem 5.2] for details.

▶ **Theorem 4.** *Let G be a finitely generated group and let Σ be a finite symmetric generating set for G. The randomized streaming space complexity of $\mathsf{WP}(G, \Sigma)$ is $\Omega(\log \log \gamma_{G,\Sigma}(\lfloor n/2 \rfloor))$.*

▶ Remark 5. Since every f.g. infinite group has growth at least n, Theorem 4 has the following consequence: If G is a f.g. infinite group, then the randomized streaming space complexity of G is $\Omega(\log \log n)$.

▶ Remark 6. Later in this paper, we will make use of the following two famous results on the growth of groups, see also [9, 28]:

- Gromov's theorem [16]: A f.g. group has polynomial growth iff it is virtually nilpotent.

- Wolf-Milnor theorem [30, 42]; see also [9, p. 202]: A f.g. solvable group G is either virtually nilpotent (and hence has polynomial growth) or there is a constant $c > 1$ such that G has growth c^n (i.e., G has exponential growth). It is well known that the same dichotomy also holds for f.g. linear groups. This is a consequence of Tits alternative [37]: A f.g. linear group G is either virtually solvable or contains a free group of rank at least two (in which case G has exponential growth).

The dichotomy theorem of Milnor and Wolf does not generalize to all f.g. groups. Grigorchuk [14] constructed a f.g.group whose growth is lower bounded by $\exp(n^{0.515})$ [6] and upper bounded by $\exp(n^{0.768})$ [5]. The streaming space complexity of this remarkable group will be studied in Theorem 23.

5 Injective randomized streaming algorithms

For a semi-probabilistic finite automaton $\mathcal{A} = (Q, \Sigma, \iota, \rho, F)$ and some boolean condition $\mathcal{E}(q)$ that depends on the state $q \in Q$, we define the probability

$$\mathop{\mathsf{Prob}}_{q \in Q}[\mathcal{E}(q)] = \sum_{q \in Q, \mathcal{E}(q)=1} \iota(q).$$

Let G be a f.g. group G with the finite generating set Σ. A randomized streaming algorithm $(\mathcal{A}_n)_{n \geq 0}$ with $\mathcal{A}_n = (Q_n, \Sigma, \iota_n, \rho_n, F_n)$ is called ϵ-*injective* for G (with respect to Σ) if the following properties hold for all $n \geq 0$ and all words $u, v \in \Sigma^{\leq n}$:

- $\mathcal{A}_n$ is semi-randomized.
- If $u \equiv_G v$ then $\mathsf{Prob}_{q \in Q_n}[\rho_n(q, u) = \rho_n(q, v)] \geq 1 - \epsilon$.
- If $u \not\equiv_G v$ then $\mathsf{Prob}_{q \in Q_n}[\rho_n(q, u) \neq \rho_n(q, v)] \geq 1 - \epsilon$.

Note that the set F_n of final states of $\mathcal{A}_n$ is not important and we will just write $\mathcal{A}_n = (Q_n, \Sigma, \iota_n, \rho_n)$ in the following if we talk about an ϵ-injective randomized streaming algorithm $(\mathcal{A}_n)_{n \geq 0}$. The easy proof of the following lemma can be found in [25, Lemma 6.1].

▶ **Lemma 7.** *If $\mathcal{R}$ is an ϵ-injective randomized streaming algorithm for G w.r.t. Σ, then $\mathsf{WP}(G, \Sigma)$ has an ϵ-correct semi-randomized streaming algorithm with space complexity $2 \cdot s(\mathcal{R}, n)$.*

Due to Lemma 7, our goal in the rest of the paper will the construction of space efficient ϵ-injective randomized streaming algorithms for groups. We will need ϵ-injectivity for wreath products and free products; see Sections 7.2 and 7.3.

6 Randomized streaming algorithms for linear groups

For every f.g. linear group, the word problem can be solved in logarithmic space. This was shown by Lipton and Zalcstein [23] (if the underlying field has characteristic zero) and Simon [36] (if the underlying field has prime characteristic). The idea is to carry out all computations modulo sufficiently many small prime numbers. This idea can be easily turned into a randomized streaming algorithm by randomly choosing a small prime number. With some care, one can turn their idea into an $\epsilon(n)$-injective randomized streaming algorithm with $\epsilon(n) = 1/n^c$ for a constant c and space complexity $\mathcal{O}(\log n)$; see [25, Theorem 7.1] for details.

▶ **Theorem 8.** *For every f.g. linear group G and every $c > 0$ there exists an $\epsilon(n)$-injective randomized streaming algorithm with $\epsilon(n) = 1/n^c$ and space complexity $\mathcal{O}(\log n)$.*

Every nilpotent group is linear. For nilpotent groups we can improve the algorithm from the proof of Theorem 8, at least if we sacrifice the inverse polynomial error probability. The proof of the following theorem (see [25, Theorem 7.2]) uses the fact that every f.g. nilpotent group is a finite extension of a nilpotent group that can be embedded in the group $\mathsf{UT}_d(\mathbb{Z})$ of d-dimensional unitriangular matrices over $\mathbb{Z}$. The entries in a product of n such matrices $A_1, \ldots, A_n$ can be bounded by $\mathcal{O}(n^{d-1})$, provided all entries in the A_i are of size $\mathcal{O}(1)$ (in absolute value). This allows to compute modulo a random prime number with $\mathcal{O}(\log \log n)$ bits.

▶ **Theorem 9.** *For every f.g. nilpotent group G and every constant $c > 0$ there exists an $\epsilon(n)$-injective randomized streaming algorithm with $\epsilon(n) = 1/\log^c n$ and space complexity $\mathcal{O}(\log \log n)$.*

Note that if G is infinite, the upper bound from Theorem 9 is sharp up to constant factors even if we allow a constant error probability; see Remark 5.

7 Closure properties for streaming space complexity

In this section, we will show that many group theoretical constructions preserve randomized streaming space complexity.

7.1 Easy cases: finite extensions and direct products

For many algorithmic problems in group theory, the complexity is preserved by finite extensions. This is also true for the streaming space complexity of the word problem; see also [25, Theorem 8.1]:

▶ **Theorem 10.** *Assume that H is a f.g. group and G is a subgroup of H of finite index (hence, also G must be finitely generated). Assume that $\mathcal{R}$ is an ϵ-injective randomized streaming algorithm for G. Then H has an ϵ-injective randomized streaming algorithm with space complexity $s(\mathcal{R}, c \cdot n) + \mathcal{O}(1)$ for some constant c.*

Recall that Gromov proved that a finitely generated group has polynomial growth if and only if it is virtually nilpotent.

▶ **Corollary 11.** *Let G be an infinite finitely generated linear group.*

- *If G is virtually nilpotent then the randomized streaming space complexity of G is $\Theta(\log \log n)$.*
- *If G is not virtually nilpotent then the randomized streaming space complexity of G is $\Theta(\log n)$.*

Proof. The upper bounds follow from Theorems 8 and 9. Since G is infinite, the randomized streaming space complexity of the word problem of G is $\Omega(\log \log n)$ (see Remark 5), which yields the lower bound for the virtually nilpotent case. If G is not virtually nilpotent, then G has growth c^n for some constant $c > 1$ (see Remark 6), which yields the lower bound $\Theta(\log n)$ by Theorem 4. ◀

It is conjectured that for every f.g. group G that is not virtually nilpotent the growth is lower bounded by $\exp(n^{0.5})$. This is known as the *gap conjecture* [15]. It would imply that for every f.g. group that is not virtually nilpotent the randomized streaming space complexity is lower bounded by $\Omega(\log n)$.

Also direct products preserve the streaming space complexity of the word problem (simply run the streaming algorithms for the two factor groups in parallel):

▶ **Lemma 12.** *Let G and H be f.g. groups for which there exist $\epsilon(n)$-injective randomized streaming algorithms $\mathcal{R}$ and $\mathcal{S}$, respectively. Then there exists a $2\epsilon(n)$-injective randomized streaming algorithm for $G \times H$ with space complexity $s(\mathcal{R}, n) + s(\mathcal{S}, n)$.*

Recall that a group G is metabelian if it has an abelian normal subgroup $A \leq G$ such that the quotient G/A is abelian as well. Every f.g. metabelian group can be embedded into a direct product of linear groups (over fields of different characteristics) [40]. Hence, with Lemma 12 and Theorem 8 we obtain:

▶ **Corollary 13.** *For every f.g. metabelian group and every $c > 0$ there exists an $\epsilon(n)$-injective randomized streaming algorithm with $\epsilon(n) = 1/n^c$ and space complexity $\mathcal{O}(\log n)$.*

7.2 Randomized streaming algorithms for wreath products

Let G and H be groups. Consider the direct sum $K = \bigoplus_{g \in G} H_g$, where H_g is a copy of H. We view K as the set $H^{(G)}$ of all mappings $f \colon G \to H$ such that $\mathsf{supp}(f) := \{g \in G \mid f(g) \neq 1\}$ is finite, together with pointwise multiplication in H as the group operation. The set $\mathsf{supp}(f) \subseteq G$ is called the *support* of f. The group G has a natural left action on $H^{(G)}$ given by $gf(a) = f(g^{-1}a)$, where $f \in H^{(G)}$ and $g, a \in G$. The corresponding semidirect product $H^{(G)} \rtimes G$ is the (restricted) *wreath product* $H \wr G$. In other words:

- Elements of $H \wr G$ are pairs (f, g), where $g \in G$ and $f \in H^{(G)}$.
- The multiplication in $H \wr G$ is defined as follows: Let $(f_1, g_1), (f_2, g_2) \in H \wr G$. Then $(f_1, g_1)(f_2, g_2) = (f, g_1 g_2)$, where $f(a) = f_1(a) f_2(g_1^{-1} a)$ for all $a \in G$.

Intuitively, the mapping $a \mapsto f_2(g_1^{-1}a)$ is the mapping f_2 shifted by g_1.

Clearly, G is a subgroup of $H \wr G$. We also regard H as a subgroup of $H \wr G$ by identifying H with the set of all $f \in H^{(G)}$ with $\mathsf{supp}(f) \subseteq \{1\}$. This copy of H together with G generates $H \wr G$. In particular, if $G = \langle \Sigma \rangle$ and $H = \langle \Gamma \rangle$ with $\Sigma \cap \Gamma = \emptyset$ then $H \wr G$ is generated by $\Sigma \cup \Gamma$. In [32] it was shown that the word problem of a wreath product $H \wr G$ is TC^0-reducible to the word problems for G and H.

In this section, we study streaming algorithms for wreath products. The case of a wreath product $H \wr G$ with G finite is easy:

▶ **Proposition 14.** *Let H be a f.g. group for which there exists an ϵ-injective randomized streaming algorithm $\mathcal{R} = (\mathcal{A}_n)_{n \geq 0}$ and let G be a finite group of size c. Then, there exists an $(c \cdot \epsilon)$-injective randomized streaming algorithm for $H \wr G$ with space complexity $\mathcal{O}(s(\mathcal{R}, n))$.*

Proof. We run c independent copies of $\mathcal{A}_n$ (for the direct product of c copies of H). In addition we have to store an element of G. ◀

The case of a wreath product $H \wr G$ with G infinite turns out to be more interesting. In Section 8 we will prove a lower bound for the case that H is non-abelian. In this section, we consider the case where H is abelian. Our construction will start with ϵ-injective randomized streaming algorithms for G and uses the following simple fact.

▶ **Lemma 15.** *Let $(\mathcal{A}_n)_{n \geq 0}$ be an ϵ-injective randomized streaming algorithm for the finitely generated group G with respect to the generating set Σ. Let $\mathcal{A}_n = (Q_n, \Sigma, \iota, \rho)$. Consider a set $S \subseteq \Sigma^{\leq n}$. For every state q of $\mathcal{A}_n$ consider the equivalence relation $\equiv_q$ on S with $u \equiv_q v$ if and only if $\rho_n(q, u) = \rho_n(q, v)$. Then*

$$\mathop{\mathsf{Prob}}_{q \in Q_n}[\equiv_q \text{ coincides with } \equiv_G \text{ on } S] \geq 1 - \epsilon \cdot \binom{|S|}{2}.$$

Proof. For all $u, v \in S$, the probability that either $u \equiv_G v$ and $u \not\equiv_q v$ or $u \not\equiv_G v$ and $u \equiv_q v$ is bounded by ϵ. The lemma follows from the union bound since there are $\binom{|S|}{2}$ many unordered pairs. ◀

We start with the case of a wreath product $\mathbb{Z} \wr G$. Note that the following theorem makes only sense if $\epsilon < 1/2n^2$. On the other hand, such an inverse polynomial error probability can be achieved for linear groups by Theorem 8.

▶ **Theorem 16.** *Let G be a f.g. infinite group and $\mathcal{R} = (\mathcal{A}_n)_{n\geq 0}$ an ϵ-injective randomized streaming algorithm for G. Let d be a fixed constant and $\zeta = 2\epsilon n^2 + \max\{\epsilon, 1/n^d\}$. Then there exists an ζ-injective randomized streaming algorithm $\mathcal{S} = (\mathcal{B}_n)_{n\geq 0}$ for $\mathbb{Z} \wr G$ with space complexity $2 \cdot s(\mathcal{R}, n) + \Theta(\log n)$.*

Proof sketch. A complete proof of the theorem can be found in [25, Theorem 8.9]. Fix a symmetric generating set Σ for G and let a be a generator of $\mathbb{Z}$. Let $\mathcal{A}_n = (Q_n, \Sigma, \iota_n, \rho_n)$. W.l.o.g. we can assume that $Q_n = [0, |Q_n| - 1]$. Fix an input length n. For an input word $w \in (\Sigma \cup \{a, a^{-1}\})^{\leq n}$ our randomized streaming algorithm for $\mathbb{Z} \wr G$ runs $\mathcal{A}_n$ on the projection $\pi_\Sigma(w)$ of the word w to the subalphabet Σ. Initially, the algorithm guesses a state $q \in Q_n$ according to the initial state distribution ι_n and (independently from q) a prime number $p \in [2, \alpha]$. The number α will be fixed latter. For the moment, let us only mention that p will have at most $s(\mathcal{R}, n) + \Theta(\log n)$ bits. Apart from the current state of $\mathcal{A}_n$ the algorithm also stores a number $z \in [0, p-1]$ which initially is set to zero.

Assume that at some time instant the algorithm reads the letter a^γ, where $\gamma \in \{-1, 1\}$. Let q be the current $\mathcal{A}_n$-state, which is a number in the range $[0, |Q_n| - 1]$. The above prime number p has to be chosen uniformly from the set of all primes of size $\Theta(|Q_n| \cdot n^{d+1})$. Then the algorithm updates $z \in [0, p-1]$ as follows:

$$z := (z + \gamma \cdot (n+1)^q) \bmod p.$$

With our input word w and a state q of $\mathcal{A}_n$ we associate a polynomial $P_{q,w}(x) \in \mathbb{Z}[x]$ as follows: Let R_w be the set of all prefixes of w that end with a letter a^γ and for $s \in R_w$ define $\sigma(s) = \gamma \in \{-1, 1\}$ if s ends with a^γ. Moreover, for every $s \in R_w$ consider the $\mathcal{A}_n$-state $q_s = \rho_n(q, \pi_\Sigma(s)) \in [0, |Q_n| - 1]$. We then define the polynomial

$$P_{q,w}(x) := \sum_{s \in R_w} \sigma(s) \cdot x^{q_s}.$$

Note that this polynomial has degree at most $|Q_n| - 1$ and all its coefficients have absolute value at most n. Moreover, the number $z = z(p, q, w)$ computed by the algorithm on input w for the random choice $p \in [2, \alpha], q \in [0, |Q_n| - 1]$ is

$$z(p, q, w) = P_{q,w}(n+1) \bmod p. \tag{1}$$

This concludes the description of the streaming algorithm. It remains to show for all words $u, v \in (\Sigma \cup \{a, a^{-1}\})^{\leq n}$ the following:

(a) If $u \equiv_{\mathbb{Z} \wr G} v$ then $\underset{p \in [2,\alpha], q \in Q_n}{\mathsf{Prob}}[\rho_n(q, \pi_\Sigma(u)) = \rho_n(q, \pi_\Sigma(v)) \wedge z(p, q, u) = z(p, q, v)] \geq 1 - \zeta$.

(b) If $u \not\equiv_{\mathbb{Z} \wr G} v$ then $\underset{p \in [2,\alpha], q \in Q_n}{\mathsf{Prob}}[\rho_n(q, \pi_\Sigma(u)) \neq \rho_n(q, \pi_\Sigma(v)) \vee z(p, q, u) \neq z(p, q, v)] \geq 1 - \zeta$.

Let $(f_u, g_u) \in \mathbb{Z} \wr G$ (resp., $(f_v, g_v) \in \mathbb{Z} \wr G$) be the group element represented by the word u (resp., v). First assume that $u \equiv_{\mathbb{Z} \wr G} v$, i.e., $f_u = f_v$ and $g_u = g_v$. From $g_u = g_v$ we get

$$\underset{q \in Q_n}{\mathsf{Prob}}[\rho_n(q, \pi_\Sigma(u)) = \rho_n(q, \pi_\Sigma(v))] \geq 1 - \epsilon. \tag{2}$$

Moreover, from $f_u = f_v$ and Lemma 15 one can deduce

$$\operatorname*{Prob}_{p\in[2,\alpha],q\in Q_n}[z(p,q,u) = z(p,q,v)] \geq \operatorname*{Prob}_{q\in Q_n}[P_{q,u}(x) = P_{q,v}(x)] \geq 1 - 2\epsilon n^2. \tag{3}$$

Finally, (2) and (3) easily yield the conclusion of point (a). Now assume that $u \not\equiv_{\mathbb{Z}\wr G} v$. If $g_u \neq g_v$, i.e., then we get

$$\operatorname*{Prob}_{q\in Q_n}[\rho_n(q,\pi_\Sigma(u)) \neq \rho_n(q,\pi_\Sigma(v))] \geq 1 - \epsilon \geq 1 - \zeta.$$

On the other hand, if the mappings f_u and f_v differ, then from Lemma 15 we obtain

$$\operatorname*{Prob}_{q\in Q_n}[P_{q,u}(n+1) \neq P_{q,v}(n+1)] \geq \operatorname*{Prob}_{q\in Q_n}[P_{q,u}(x) \neq P_{q,v}(x)] \geq 1 - 2\epsilon n^2.$$

The first inequality follows from Cauchy's bound. This, together with (1) and some standard bounds on the number of different prime factors of $|P_{q,u}(n+1) - P_{q,v}(n+1)|$ yields $\operatorname{Prob}_{p\in[2,\alpha],q\in Q_n}[z(p,q,u) \neq z(p,q,v)] \geq 1 - \zeta$ and finally the conclusion of point (b). ◀

It is easy to extend Theorem 16 to a wreath product $\mathbb{Z}_p \wr G$ with p prime. For a wreath product $\mathbb{Z}_{p^k} \wr G$ with p a prime and $k \geq 2$ we can only prove the following weaker statement using a polynomial identity testing algorithm for local rings from [3]. Putting it all together we can show the following result; see [25, Section 8.3] for details.

▶ **Theorem 17.** *Let G be a f.g. group for which there exists an ϵ-injective randomized streaming algorithm $\mathcal{R}$. Let A be a finitely generated abelian group. Then for all constants $\epsilon \leq \epsilon' < 1$ and $d \geq 1$ there exists a ζ-injective randomized streaming algorithm $\mathcal{S}$ for $A \wr G$ with the following properties:*

- *$\zeta \leq \mathcal{O}(\epsilon' + \epsilon n^2)$ and $s(\mathcal{S},n) \leq \mathcal{O}(s(\mathcal{R},n)^2 + \log n)$*
- *$\zeta \leq \mathcal{O}(1/n^d + \epsilon n^2)$ and $s(\mathcal{S},n) \leq \mathcal{O}(s(\mathcal{R},n) + \log n)$ if A is a direct product of copies of $\mathbb{Z}$ and $\mathbb{Z}_p$ with p prime.*

▶ **Corollary 18.** *Every free solvable group has randomized streaming space complexity $\Theta(\log n)$.*

Proof. Magnus' embedding theorem [27] says that every free solvable group can be embedded into an iterated wreath product $\mathbb{Z}^m \wr (\mathbb{Z}^m \wr (\mathbb{Z}^m \wr \cdots))$. Since $\mathbb{Z}^m$ is linear, we can, using Theorem 8, obtain an $\epsilon(n)$-injective randomized streaming algorithm for $\mathbb{Z}^m$ with space complexity $\mathcal{O}(\log n)$ for every inverse polynomial $\epsilon(n)$. We then apply the second statement of Theorem 17 a constant number of times and obtain a randomized streaming algorithm with space complexity $\mathcal{O}(\log n)$. The lower bound follows from Theorem 4 and the Milnor-Wolf theorem (see Remark 6). ◀

In [38] it is shown that the word problem of a free solvable group can be solved with a randomized algorithm running in time $\mathcal{O}(n \cdot \log^k n)$ for some constant k. Our algorithm achieves the same running time (because for every new input symbol, only numbers of bit length $\mathcal{O}(\log n)$ have to be manipulated). In contrast to our algorithm, the algorithm from [38] is non-streaming and does not work in logarithmic space.

7.3 Randomized streaming algorithms for free products

In [39], Waack proved that the word problem of a free product $G * H$ of two f.g. groups G and H can be solved in logspace if the word problems of G and H can be solved in logspace. Here, we show that Waack's reduction can be also used for randomized streaming algorithms.

For a group G and two subgroups $A, B \leq G$, the commutator subgroup $[A, B] \leq G$ is the group generated by all commutators $[a, b] = a^{-1}b^{-1}ab$ with $a \in A$ and $b \in B$. Let us denote with $F(\Sigma)$ the free group generated by the set Σ. A free group F is freely generated by the set $A \subseteq F$ if F is isomorphic to $F(A)$. It is well known that the free group $F_2 = F(\{a, b\})$ of rank 2 contains a copy of the free group $F(\mathbb{N})$ of countable infinite rank. For instance, the mapping $\phi : \mathbb{N} \to F_2$ with $\phi(n) = a^{-n}ba^n$ defines an injective homomorphism $\phi : F(\mathbb{N}) \to F_2$. The following group theoretic lemma underlies Waack's reduction:

▶ **Lemma 19** (c.f. [39, Proposition 4.2]). *Let G and H be groups. Then $[G, H]$ is a normal subgroup of the free product $G * H$ such that $(G * H)/[G, H] \cong G \times H$. Moreover, $[G, H]$ is a free group that is freely generated by the set of commutators $\{[g, h] \mid g \in G \setminus \{1\}, h \in H \setminus \{1\}\}$.*

▶ **Theorem 20.** *Let G and H be a f.g. groups for which there exist ϵ-injective randomized streaming algorithms $\mathcal{R} = (\mathcal{A}_n)_{n\geq 0}$ and $\mathcal{S} = (\mathcal{B}_n)_{n\geq 0}$, respectively. Then, there exists a $(4n^2 + 1)\epsilon$-injective randomized streaming algorithm $(\mathcal{C}_n)_{n\geq 0}$ for $G * H$ with space complexity $s(\mathcal{R}, n) + s(\mathcal{S}, n) + \mathcal{O}(\log n)$.*

The proof of Theorem 20 is a bit technical and can be found in [25, Section 8.2]. In order to test whether a word w is trivial in $G * H$, Waack checks whether the image of w in the quotient $(G * H)/[G, H] \cong G \times H$ is trivial (for which algorithms for G and H can be used). If this holds, then w represents an element of the free group $[G, H]$, which by the above remark can be embedded into F_2. Waack then computes from w the corresponding image in F_2. Basically, we follow the same strategy but only obtain the image in F_2 with high probability using Lemma 15. For F_2 (a linear group) we can then apply Theorem 8.

8 Lower bounds

In this section, we will construct groups with a large randomized streaming space complexity. We will make use of the disjointness problem from communication complexity. The *disjointness problem* is defined as follows: Alice (resp., Bob) has a bit string $u \in \{0, 1\}^n$ (resp., $v \in \{0, 1\}^n$) and their goal is to determine whether there is a position $1 \leq i \leq n$ such that $u[i] = v[i] = 1$ ($u[i]$ and $v[i]$ are the bits at position i). It is well known that the randomized communication complexity for the disjointness problem is $\Theta(n)$, see e.g. [21, Section 4.6].

▶ **Theorem 21.** *Let H be a f.g. non-abelian group and G be a f.g. infinite group. The randomized streaming space complexity of $H \wr G$ is $\Theta(n)$.*

Proof. Let $\mathcal{R} = (\mathcal{A}_n)_{n\geq 0}$ be a randomized streaming algorithm for the word problem of $H \wr G$. We show that we obtain a randomized communication protocol for the disjointness problem with communication cost $3 \cdot s(\mathcal{R}, 12n - 8)$.

Fix $n \geq 1$ and two elements $g, h \in H$ with $[g, h] \neq 1$. We can assume that g and h are generators of H. We also fix a finite generating set for G. Let $s := t_1 t_2 \cdots t_{n-1}$ be a word over the generators of G such that $t_1 t_2 \cdots t_i$ and $t_1 t_2 \cdots t_j$ represent different elements whenever $i \neq j$. Such a word exists since the Cayley graph of G is an infinite locally finite graph and hence contains an infinite ray. For a word $w = a_0 a_1 \cdots a_{n-1} \in \{0, 1\}^n$ and an element $x \in \{g, h, g^{-1}, h^{-1}\}$ define the word $w[x] = x^{a_0} t_1 x^{a_1} t_2 \cdots x^{a_{n-2}} t_{n-1} x^{a_{n-1}} s^{-1}$. It represents the element $(f_{w,x}, 1) \in H \wr G$ with $\mathsf{supp}(f_{w,x}) = \{t_1 \dots t_i \mid i \in [0, n-1], w[i] = 1\}$ and $f_{w,x}(t) = x$ for all $t \in \mathsf{supp}(f_{w,x})$. Therefore, for two words $u, v \in \{0, 1\}^n$ we have $u[g]v[h]u[g^{-1}]v[h^{-1}] = 1$ in $H \wr G$ if and only if there is a position $i \in [0, n-1]$ with $u[i] = v[i] = 1$. Note that the length of the word $u[g]v[h]u[g^{-1}]v[h^{-1}]$ is $4(3n - 2) = 12n - 8$.

Our randomized communication protocol for the disjointness problem works as follows, where $u \in \{0, 1\}^n$ is the input for Alice and $v \in \{0, 1\}^n$ is the input for Bob.

- Alice reads the word $u[g]$ into $\mathcal{A}_{12n-8}$ and sends the resulting state to Bob.
- Bob continues the run in the state he received from Alice, reads the word $v[h]$ into the automaton and sends the resulting state back to Alice.
- Alice continues the run with the word $u[g^{-1}]$ and sends the resulting state of Bob.
- Bob continues the run with $v[h^{-1}]$ and finally accepts if the resulting state is accepting.

Both Alice and Bob use their random coins in order to make the random decisions in the PFA $\mathcal{A}_{12n-8}$. Clearly, the protocol is correct and its communication cost is $3 \cdot s(\mathcal{R}, 12n-8)$. We hence must have $3 \cdot s(\mathcal{R}, 12n-8) \in \Omega(n)$ which implies $s(\mathcal{R}, m) \in \Omega(m)$. ◀

In 1965, Thompson introduced three finitely presented groups $F < T < V$ acting on the unit-interval, the unit-circle and the Cantor set, respectively. Of these three groups, F received most attention (the reader should not confuse F with a free group). This is mainly due to the still open conjecture that F is not amenable. The group F consists of all homeomorphisms of the unit interval that are piecewise affine, with slopes a power of 2 and dyadic breakpoints. It is a finitely presented group. Important for us is the fact that F contains a copy of $F \wr \mathbb{Z}$ [17, Lemma 20]. Since F is non-abelian, Theorem 21 implies:

▶ **Corollary 22.** *The randomized streaming space complexity of Thompson's group F is $\Theta(n)$.*

Grigorchuk's group was introduced by Grigorchuk in [14]. It is a f.g. group of automorphisms of the infinite binary tree; the generators are usually denoted a, b, c, d and satisfy the identities $a^2 = b^2 = c^2 = d^2 = 1$ and $bc = cb = d, bd = db = c, dc = cd = b$. Grigorchuk's group is a f.g. infinite torsion group and was the first example of a group with intermediate growth and the first example of a group that is amenable but not elementary amenable.

▶ **Theorem 23.** *Let G be the Grigorchuk group. Then the following hold:*
- *The deterministic streaming space complexity of G is $\mathcal{O}(n^{0.768})$.*
- *The randomized streaming space complexity of G is $\Omega(n^{0.5})$.*

Proof sketch. The first statement follows from Theorem 3 and the upper growth bound $\exp(n^{0.768})$ for the Grigorchuk group; see [5]. For the second statement we use a non-abelian subgroup $K \leq G$ such that K contains a copy of $K \times K$, see [9, p. 262]. This allows a similar reduction from the disjointness problem as in the proof of Theorem 21; see [25, Theorem 9.3] for details. ◀

9 Open problems

We conclude with some open problems.
- Can the space bound $\mathcal{O}(s(\mathcal{R}, n)^2 + \log n)$ in Theorem 17 be reduced to $\mathcal{O}(s(\mathcal{R}, n) + \log n)$?
- What is the space complexity of randomized streaming algorithms for hyperbolic groups? The best complexity bound for the word problem for a hyperbolic group is LogCFL, which is contained in $\mathsf{DSPACE}(\log^2 n)$ [24].
- Is there a group that is not residually finite and for which there exists a randomized streaming algorithm with space complexity $o(n)$? An example of group that is not residually finite is the Baumslag-Solitar group $\mathsf{BS}(2, 3)$. The word problem for every Baumslag-Solitar group $\mathsf{BS}(p, q)$ can be solved in logarithmic space [41].

References

1 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999. doi:10.1006/jcss.1997.1545.

2 Anatolij W. Anissimov and Franz D. Seifert. Zur algebraischen Charakteristik der durch kontext-freie Sprachen definierten Gruppen. *Elektronische Informationsverarbeitung und Kybernetik*, 11(10–12):695–702, 1975.

3 Vikraman Arvind, Partha Mukhopadhyay, and Srikanth Srinivasan. New results on noncommutative and commutative polynomial identity testing. *Computational Complexity*, 19(4):521–558, 2010. doi:10.1007/s00037-010-0299-8.

4 Ajesh Babu, Nutan Limaye, Jaikumar Radhakrishnan, and Girish Varma. Streaming algorithms for language recognition problems. *Theoretical Computer Science*, 494:13–23, 2013. doi:10.1016/j.tcs.2012.12.028.

5 Laurent Bartholdi. The growth of Grigorchuk's torsion group. *International Mathematics Research Notices*, 20:1049–1054, 1998. doi:10.1155/S1073792898000622.

6 Laurent Bartholdi. Lower bounds on the growth of a group acting on the binary rooted tree. *International Journal of Algebra and Computation*, 11(01):73–88, 2001. doi:10.1142/S0218196701000395.

7 Gabriel Bathie and Tatiana Starikovskaya. Property testing of regular languages with applications to streaming property testing of visibly pushdown languages. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming, ICALP 2021*, volume 198 of *LIPIcs*, pages 119:1–119:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.119.

8 John W. Cannon, William J. Floyd, and Walter R. Parry. Introductory notes on Richard Thompson's groups. *L'Enseignement Mathématique*, 42(3):215–256, 1996.

9 Pierre de la Harpe. *Topics in Geometric Group Theory*. University of Chicago Press, 2000.

10 Max Dehn. Über unendliche diskontinuierliche Gruppen. *Mathematische Annalen*, 71:116–144, 1911. In German. doi:10.1007/BF01456932.

11 Nathanaël Fijalkow. The online space complexity of probabilistic languages. In *Proceedings of the International Symposium on Logical Foundations of Computer Science, LFCS 2016*, volume 9537 of *Lecture Notes in Computer Science*, pages 106–116. Springer, 2016. doi:10.1007/978-3-319-27683-0_8.

12 Nathanaël François, Frédéric Magniez, Michel de Rougemont, and Olivier Serre. Streaming property testing of visibly pushdown languages. In *Proceedings of the 24th Annual European Symposium on Algorithms, ESA 2016*, volume 57 of *LIPIcs*, pages 43:1–43:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ESA.2016.43.

13 William Timothy Gowers and Emanuele Viola. Interleaved group products. *SIAM Journal on Computing*, 48(2):554–580, 2019. doi:10.1137/17M1126783.

14 Rostislav I. Grigorchuk. Burnside's problem on periodic groups. *Functional Analysis and Its Applications*, 14:41–43, 1980. doi:10.1007/BF01078416.

15 Rostislav I. Grigorchuk. On the gap conjecture concerning group growth. *Bulletin of Mathematical Sciences*, 4(1):113–128, 2014. doi:10.1007/s13373-012-0029-4.

16 Mikhail Gromov. Groups of polynomial growth and expanding maps. *Publications Mathématiques de L'Institut des Hautes Scientifiques*, 53:53–78, 1981. doi:10.1007/BF02698687.

17 Victor S. Guba and Mark V. Sapir. On subgroups of the R. Thompson group F and other diagram groups. *Matematicheskii Sbornik*, 190(8):3–60, 1999. doi:10.1070/SM1999v190n08ABEH000419.

18 Derek F. Holt, Sarah Rees, and Claas E. Röver. *Groups, Languages and Automata*, volume 88 of *London Mathematical Society Student Texts*. Cambridge University Press, 2017. doi:10.1017/9781316588246.

19 Richard M. Karp. Some bounds on the storage requirements of sequential machines and Turing machines. *Journal of the Association for Computing Machinery*, 14(3):478–489, 1967. doi:10.1145/321406.321410.

20 Daniel König and Markus Lohrey. Evaluation of circuits over nilpotent and polycyclic groups. *Algorithmica*, 80(5):1459–1492, 2018. doi:10.1007/s00453-017-0343-z.

21 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997. doi:10.1017/CBO9780511574948.

22 Jörg Lehnert and Pascal Schweitzer. The co-word problem for the Higman-Thompson group is context-free. *Bulletin of the London Mathematical Society*, 39(2):235–241, February 2007. doi:10.1112/blms/bdl043.

23 Richard J. Lipton and Yechezkel Zalcstein. Word problems solvable in logspace. *Journal of the Association for Computing Machinery*, 24(3):522–526, 1977. doi:10.1145/322017.322031.

24 Markus Lohrey. Decidability and complexity in automatic monoids. *International Journal of Foundations of Computer Science*, 16(4):707–722, 2005. doi:10.1142/S0129054105003248.

25 Markus Lohrey and Lukas Lück. Streaming word problems. *CoRR*, abs/2202.04060, 2022. doi:10.48550/ARXIV.2202.04060.

26 Frédéric Magniez, Claire Mathieu, and Ashwin Nayak. Recognizing well-parenthesized expressions in the streaming model. *SIAM Journal on Computing*, 43(6):1880–1905, 2014. doi:10.1137/130926122.

27 Wilhelm Magnus. On a theorem of Marshall Hall. *Annals of Mathematics. Second Series*, 40:764–768, 1939. doi:10.2307/1968892.

28 Avinoam Mann. *How Groups Grow*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2011. doi:10.1017/CBO9781139095129.

29 Charles F Miller III. Decision problems for groups – survey and reflections. In G. Baumslag and Charles F Miller III, editors, *Algorithms and classification in combinatorial group theory*, pages 1–59. Springer, 1992. doi:10.1007/978-1-4613-9730-4_1.

30 John Milnor. Growth of finitely generated solvable groups. *Journal of Differential Geometry*, 2(4):447–449, 1968. doi:10.4310/jdg/1214428659.

31 David E. Muller and Paul E. Schupp. Groups, the theory of ends, and context-free languages. *Journal of Computer and System Sciences*, 26:295–310, 1983. doi:10.1016/0022-0000(83)90003-X.

32 Alexei Myasnikov, Vitaly Roman'kov, Alexander Ushakov, and AnatolyVershik. The word and geodesic problems in free solvable groups. *Transactions of the American Mathematical Society*, 362(9):4655–4682, 2010. doi:10.1090/S0002-9947-10-04959-7.

33 Azaria Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971. doi:10.1016/C2013-0-11297-4.

34 Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963. doi:10.1016/S0019-9958(63)90290-0.

35 Jeffrey Shallit and Yuri Breitbart. Automaticity I: properties of a measure of descriptional complexity. *Journal of Computer and System Sciences*, 53(1):10–25, 1996. doi:10.1006/jcss.1996.0046.

36 Hans-Ulrich Simon. Word problems for groups and contextfree recognition. In *Proceedings of Fundamentals of Computation Theory, FCT 1979*, pages 417–422. Akademie-Verlag, 1979.

37 Jacques Tits. Free subgroups in linear groups. *Journal of Algebra*, 20:250–270, 1972. doi:10.1016/0021-8693(72)90058-0.

38 Alexander Ushakov. Algorithmic theory of free solvable groups: Randomized computations. *Journal of Algebra*, 407:178–200, 2014. doi:10.1016/j.jalgebra.2014.02.014.

39 Stephan Waack. The parallel complexity of some constructions in combinatorial group theory. *Journal of Information Processing and Cybernetics, EIK*, 26:265–281, 1990.

40 Bertram A. F. Wehrfritz. On finitely generated soluble linear groups. *Mathematische Zeitschrift*, 170:155–167, 1980. doi:10.1007/BF01214771.

41 Armin Weiß. A logspace solution to the word and conjugacy problem of generalized Baumslag-Solitar groups. In *Algebra and Computer Science*, volume 677 of *Contemporary Mathematics*. American Mathematical Society, 2016. doi:10.1090/conm/677.

42 Joseph A. Wolf. Growth of finitely generated solvable groups and curvature of Riemannian manifolds. *Journal of Differential Geometry*, 2(4):421–446, 1968. doi:10.4310/jdg/1214428658.