Logspace computations in graph groups and Coxeter groups

Volker Diekert[1], Jonathan Kausch[1], and Markus Lohrey[2]

[1] FMI, Universität Stuttgart, Germany
[2] Insitut für Informatik, Universität Leipzig, Germany

Abstract. Computing normal forms in groups (or monoids) is in general harder than solving the word problem (equality testing). However, normal form computation has a much wider range of applications. It is therefore interesting to investigate the complexity of computing normal forms for important classes of groups. We show that shortlex normal forms in graph groups and in right-angled Coxeter groups can be computed in logspace. Graph groups are also known as free partially commutative groups or as right-angled Artin groups in the literature. (Artin groups can be realized as subgroups of Coxeter groups.) Graph groups arise in many areas and have a close connection to concurrency theory. The connection is used here. Indeed, for our result we use a representation of group elements by Mazurkiewicz traces. These are directed acyclic node-labelled graphs (i.e. pomsets). They form an algebraic model to describe runs of concurrent systems. Concurrent systems which are deterministic and co-deterministic can be studied via inverse monoids. As an application of our results we show that the word problem for free partially commutative inverse monoids is in logspace. This result generalizes a result of Ondrusch and the third author on free inverse monoids.
All Coxeter groups are linear, so the word problem can be solved in logspace, but it is open (in the non-right-angled case) whether shortlex normal forms can be computed in logspace, or, less demanding, whether they can be computed efficiently in parallel. We show that for all Coxeter groups the set of letters occurring in the shortlex normal form of an element can be computed in logspace.

1 Introduction

The study of group-theoretic decision problems, like the word problem (Does a given word equal 1 in the group?), the conjugacy problem (Are two given words conjugated in the group?), and the isomorphism problem (Do two given group presentations yield isomorphic groups?), is a classical topic in combinatorial group theory with a long history dating back to the beginning of the 20th century, see the survey [25] for more details.[1]

With the emergence of computational complexity theory, the complexity of these decision problems in various classes of groups has developed into an active

[1] All groups in this paper are assumed to be finitely generated.

research area, where algebraic methods as well as computer science techniques complement one another in a fruitful way.

In this paper we are interested in group-theoretic problems which can be solved efficiently in parallel (hence below P). More precisely, we are interested in *deterministic logspace*, called simply *logspace* in the following. A fundamental result in this context (which is crucial in this paper, too) was shown in [22, 30]: The word problem of finitely generated linear groups belongs to logspace. (In [22], Lipton and Zalcstein proved this result for fields of characteristic 0, only.) The class of groups with a word problem in logspace is further investigated in [32]. Another important result is Cai's NC^2 algorithm for the word problem of a hyperbolic group [6]. In [23] this result was improved to LOGCFL.

Often, it is not enough to solve the word problem, but one has to compute a normal form for a given group element. Fix a finite generating set Γ (w.l.o.g. closed under inverses) for the group G. Then, a *geodesic* for $g \in G$ is a shortest word over Γ that represents g. By choosing the lexicographical smallest (w.r.t. a fixed ordering on Γ) word among all geodesics for g, one obtains the *shortlex normal form* of g. The problem of computing geodesics and various related problems were studied in [14, 15, 17, 27, 29]. It turned out that there are groups with an easy word problem (in logspace), but where simple questions related to geodesics are computationally hard. For instance, every metabelian group embeds (effectively) into a direct product of linear groups; hence its word problem can be solved in logspace. On the other hand, it is shown in [14], that the question whether a given element x of the wreath product $\mathbb{Z}/2\mathbb{Z} \wr (\mathbb{Z} \times \mathbb{Z})$ (a metabelian group) has geodesic length at most n is NP-complete. A corresponding result was shown in [27] for the free metabelian group of rank 2. Clearly, these results show that in general one cannot compute shortlex normal forms in metabelian groups in polynomial time (unless $\mathsf{P} = \mathsf{NP}$). On the positive side, for *shortlex automatic groups* [18] (i.e., automatic groups, where the underlying regular set of representatives is the set of shortlex normal forms) shortlex normal forms can be computed in quadratic time. In [27], it is also noted that geodesics in nilpotent groups can be computed in polynomial time.

In this paper, we deal with the problem of computing geodesics and shortlex normal forms in logspace. A function can be computed in logspace, if it can be computed by a deterministic *logspace transducer*. The latter is a Turing machine with three tapes: (i) a read-only input tape, (ii) a read/write work tape of length $\mathcal{O}(\log n)$, and (iii) a write-only output tape. The output is written sequentially from left to right onto the output tape. Every logspace transducer can be transformed into an equivalent deterministic polynomial time algorithm. Still better, it can be performed by a Boolean circuit of polynomial size and $\mathcal{O}(\log^2 n)$ depth. Although it is not completely obvious, the class of logspace computable functions is closed under composition. (See e.g. the textbook [28] for these facts.)

Recently, the class of groups, where geodesics and shortlex normal forms can be computed in logspace, attracted attention, see [16], where it was noted among other results that shortlex normal forms in free groups can be computed in logspace. (Implicitly, this result was also shown in [24].) In this paper, we prove

a generalization of this result. Our main result states that shortlex normal forms can be computed in logspace for *graph groups* and *right-angled Coxeter groups* (Thm. 1). Graph groups are also known as *free partially commutative groups* or as *right-angled Artin groups*. A graph group is defined by a finite undirected graph (Σ, I) by taking Σ as the set of group generators and adding the defining relation $ab = ba$ for all edges $(a, b) \in I$. Graph groups received in recent years a lot of attention in group theory because of their rich subgroup structure [2, 9, 19]. On the algorithmic side, (un)decidability results were obtained for many important group-theoretic decision problems in graph groups [8, 13]. Right-angled Coxeter groups arise from graph groups by adding all relations $a^2 = 1$ for $a \in \Sigma$. They form an important subclass of *Coxeter groups*, which are discrete reflection groups [3]. Every Coxeter group is linear and therefore has a logspace word problem [3, 10]. Moreover, there is a standard embedding of a graph group into a right-angled Coxeter group [20]. Hence, also graph groups are linear and have logspace word problems.

The computation of shortlex normal forms in Coxeter groups can be done in quadratic time, since Coxeter groups are also known to be shortlex automatic, see [5, 7]. However, no efficient parallel algorithms are known so far. In order to show that for right-angled Coxeter groups normal forms can be computed efficiently in parallel, we prove a stronger result: The computation is possible in logspace. This is an optimal result in the sense that logspace is the smallest known complexity class for the word problem in free groups; this in turn is a lower bound for our problem. We use techniques from the theory of Mazurkiewicz traces [11]. More precisely, we describe right-angled Coxeter groups by strongly confluent length-reducing trace rewriting systems. Moreover, using the geometric representation of right-angled Coxeter groups, we show that the alphabet of symbols that appear in a geodesic for g can be computed in logspace from g (Cor. 1). This alphabetic information enables us to compute shortlex normal forms in logspace. Using the special properties of the above mentioned embedding of graph groups into right-angled Coxeter groups, we can transfer our result to the former class of groups, which is the class we were interested in. For general Coxeter groups, we are still able to compute in logspace the alphabet of symbols that appear in the shortlex normal form (Thm. 2). The proof of Thm. 2 is more difficult than the proof of Cor. 1 in the sense that it uses geometry and more facts from [3]. Whether shortlex normal forms in general Coxeter groups can be computed in logspace remains open.

Finally, we apply Thm. 1 to *free partially commutative inverse monoids*. These monoids arise naturally in the context of deterministic and co-deterministic concurrent systems. This includes many real systems, because they can be viewed as deterministic concurrent systems with *undo*-operations. In [12] it was shown that the word problem for a free partially commutative inverse monoid can be solved in time $\mathcal{O}(n \log(n))$. (Decidability of the word problem is due to Da Costa [31].) Using our logspace algorithm for computing shortlex normal forms in a graph group, we can show that the word problem for a free partially commutative inverse monoid can be solved in logspace (Thm. 3). Again,

with state-of-the art techniques, this can be viewed as an optimal result. It also generalizes a corresponding result for free inverse monoids from [24]. Let us emphasize that in order to obtain Thm. 3 we have to be able to compute shortlex normal forms in graph groups in logspace; knowing only that the word problem is in logspace would not have been sufficient for our purposes.

Let us remark that for all our results it is crucial that the group (resp., the free partially commutative inverse monoids) is fixed and not part of the input. For instance, it is not clear whether for a given undirected graph (Σ, I) and a word w over $\Sigma \cup \Sigma^{-1}$ one can check in logspace whether $w = 1$ in the graph group defined by the graph (Σ, I).

2 Notation

All groups and monoids M in this paper are assumed to be finitely generated and they come with a surjective monoid homomorphism $\pi : \Sigma^* \to M$, where Σ is a finite set (also called an *alphabet*) and Σ^* is the free monoid over Σ. We assume that there is an involution $x \mapsto x^{-1}$ on M (as for all groups and inverse monoids)[2] and that $M = (\pi(\Sigma) \cup \pi(\Sigma)^{-1})^*$. If $\pi : \Sigma^* \to G$ is a surjective monoid homomorphism for a group G, then G becomes a factor group of the *free group* $F(\Sigma)$. Let $\overline{\Sigma}$ be a disjoint copy of Σ and $\Gamma = \Sigma \cup \overline{\Sigma}$. There is a unique extension of the natural mapping $\Sigma \to \overline{\Sigma} : a \mapsto \bar{a}$ such that Γ^* becomes a monoid with involution. (Indeed, we must satisfy $\bar{\bar{a}} = a$ and $\overline{a_1 \cdots a_n} = \bar{a}_n \cdots \bar{a}_1$.) Hence, we can lift our homomorphism π to a surjective monoid homomorphism $\pi : \Gamma^* \to M$ which respects the involution ($\pi(\bar{x}) = x^{-1}$). Elements of Γ (resp. Γ^*) are called *letters* (resp. *words*). The length of a word w is denoted by $|w|$. Given a surjective monoid homomorphism $\pi : \Sigma^* \to M$ and a linear order on Γ we can define the *geodesic length* and the *shortlex normal form* for elements in M as follows. For $x \in M$, the geodesic length $\|x\|$ is the length of a shortest word in $\pi^{-1}(x)$. The shortlex normal form of x is the lexicographical first word in the finite set $\{w \in \pi^{-1}(x) \mid \|x\| = |w|\}$.

3 Mazurkiewicz traces and graph groups

More details on Mazurkiewicz traces can be found in [11]. An *independence alphabet* is a pair (Σ, I), where Σ is a finite set (or *alphabet*) and $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric relation, called the *independence relation*. Thus, (Σ, I) is a finite undirected graph. The complementary relation $D = (\Sigma \times \Sigma) \setminus I$ is called a *dependence relation*. It is reflexive and symmetric. We extend (Σ, I) to a graph (Γ, I_Γ), where $\Gamma = \Sigma \cup \overline{\Sigma}$ with $\Sigma \cap \overline{\Sigma} = \emptyset$, and I_Γ is the minimal independence with $I \subseteq I_\Gamma$ and such that $(a, b) \in I_\Gamma$ implies $(a, \bar{b}) \in I_\Gamma$. The independence alphabet (Σ, I) defines a *free partially commutative monoid* (or

[2] An *involution* on a set Γ is a permutation $a \mapsto \bar{a}$ such that $\bar{\bar{a}} = a$. An involution of a monoid satisfies in addition $\overline{xy} = \bar{y}\,\bar{x}$.

trace monoid) $M(\Sigma, I)$ and a *free partially commutative group* $G(\Sigma, I)$ by:

$$M(\Sigma, I) = \Sigma^* / \{ab = ba \mid (a, b) \in I\},$$
$$G(\Sigma, I) = F(\Sigma) / \{ab = ba \mid (a, b) \in I\}.$$

Free partially commutative groups are also known as *right-angled Artin groups* or *graph groups*. Elements of $M(\Sigma, I)$ are called *(Mazurkiewicz) traces*. They have a unique description as *dependence graphs*, which are node-labelled acyclic graphs defined as follows. Let $u = a_1 \cdots a_n \in \Sigma^*$ be a word. The vertex set of the dependence graph $\mathrm{DG}(u)$ is $\{1, \ldots, n\}$ and vertex i is labelled with $a_i \in \Sigma$. There is an arc from vertex i to j if and only if $i < j$ and $(a_i, a_j) \in D$. Now, two words define the same trace in $M(\Sigma, I)$ if and only if their dependence graphs are isomorphic. A dependence graph is acyclic, so its transitive closure is a labelled partial order $\prec$, which can be uniquely represented by its *Hasse diagram*. There is an arc from i to j in the Hasse diagram, if $i \prec j$ and there does not exist k with $i \prec k \prec j$.

A trace $u \in M(\Sigma, I)$ is a *factor* of $v \in M(\Sigma, I)$, if $v \in M(\Sigma, I)uM(\Sigma, I)$. The set of letters occurring in a trace u is denoted by $\alpha(u)$. The independence relation I is extended to traces by letting $(u, v) \in I$, if $\alpha(u) \times \alpha(v) \subseteq I$. We also write $I(a) = \{b \in \Sigma \mid (a, b) \in I\}$. A trace u is called a *prime* if $\mathrm{DG}(u)$ has exactly one maximal element. Thus, if u is a prime, then we can write u as $u = va$ in $M(\Sigma, I)$, where $a \in \Sigma$ and $v \in M(\Sigma, I)$ are uniquely defined. Moreover, this property characterizes primes. A *prime prefix* of a trace u is a prime trace v such that $u = vx$ in $M(\Sigma, I)$ for some trace x. We will use the following simple fact.

Lemma 1. *Let (Σ, I) be a fixed independence relation. There is a logspace transducer that on input $u \in M(\Sigma, I)$ outputs a list of all prime prefixes of u.*

Proof. The prime prefixes of u correspond to the downward-closed subsets of the dependence graph $\mathrm{DG}(u)$ that have a unique maximal element. Assume that $u = a_1 a_2 \cdots a_n$ with $a_i \in \Sigma$. Our logspace transducer works in n phases. In the i-th phase it outputs the sequence of all symbols a_j $(j \leq i)$ such that there exists a path in $\mathrm{DG}(u)$ from j to i. Note that there exists a path from j to i in $\mathrm{DG}(u)$ if and only if there is such a path of length at most $|\Sigma|$. Since Σ is fixed, the existence of such a path can be checked in logspace. □

We use standard notation from the theory of rewriting systems, cf [4]. Let $M = M(\Sigma, I)$. A *trace rewriting system* is a finite set of rules $S \subseteq M \times M$. A rule is often written in the form $\ell \longrightarrow r$. The system S defines a one-step rewriting relation $\Longrightarrow_S \subseteq M \times M$ by $x \Longrightarrow_S y$ if there exist $(\ell, r) \in S$ and $u, v \in M$ with $x = u\ell v$ and $y = urv$ in M. By $\overset{*}{\Longrightarrow}_S$, we denote the reflexive and transitive closure of $\Longrightarrow_S$. The set $\mathrm{IRR}(S)$ denotes the set of traces to which no rule of S applies. If S is confluent and terminating, then for every $u \in M$ there is a unique $\widehat{u} \in \mathrm{IRR}(S)$ with $u \overset{*}{\Longrightarrow}_S \widehat{u}$, and $\mathrm{IRR}(S)$ is a set of normal forms for the quotient monoid M/S. If, in addition, S is length-reducing (i.e., $|\ell| > |r|$ for all $(\ell, r) \in S$), then $\|\pi(u)\| = |\widehat{u}|$ for the canonical homomorphism $\pi : M \to M/S$.

Example 1. The system $S_G = \{a\bar{a} \longrightarrow 1 \mid a \in \Gamma\}$ is (strongly) confluent and length-reducing over $M(\Gamma, I_\Gamma)$ [11]. The quotient monoid $M(\Gamma, I_\Gamma)/S_G$ is the graph group $G(\Sigma, I)$.

By Ex. 1 elements in graph groups have a unique description as *dependence graphs*, too. A trace belongs to $\mathrm{IRR}(S_G)$ if and only if it does not contain a factor $a\bar{a}$ for $a \in \Gamma$. In the dependence graph, this means that the Hasse diagram does not contain any arc from a vertex labeled a to a vertex labeled $\bar{a}$ with $a \in \Gamma$. Moreover, a word $u \in \Gamma^*$ represents a trace from $\mathrm{IRR}(S_G)$ if and only if u does not contain a factor of the form $av\bar{a}$ with $a \in \Gamma$ and $\alpha(v) \subseteq I(a)$.

4 Right-angled Coxeter groups

The *right-angled Coxeter group* $C(\Sigma, I)$ is generated by the finite alphabet Σ and has the defining relations $a^2 = 1$ for $a \in \Sigma$ and $(ab)^2 = 1$ (i.e. $ab = ba$) for $(a, b) \in I$. Similarly to the graph group $G(\Sigma, I)$, the right-angled Coxeter group $C(\Sigma, I)$ can be defined by a (strongly) confluent and length-reducing trace rewriting system (this time on $M(\Sigma, I)$ instead of $M(\Gamma, I_\Gamma)$). Let

$$S_C = \{a^2 \to 1 \mid a \in \Sigma\}.$$

Then S_C is indeed (strongly) confluent and length-reducing on $M(\Sigma, I)$ and the quotient $M(\Sigma, I)/S_C$ is $C(\Sigma, I)$. Hence we have two closely related (strongly) confluent and length-reducing trace rewriting systems: S_G defines the graph group $G(\Sigma, I)$ and S_C defines the right-angled Coxeter group $C(\Sigma, I)$. Both systems define unique normal forms of geodesic length: $\widehat{u} \in M(\Gamma, I_\Gamma)$ for S_G and $\widehat{u} \in M(\Sigma, I)$ for S_C. Note that there are no explicit commutation rules as they are *built-in* in trace theory. There is a linear time algorithm for computing $\widehat{u}$; see [11] for a more general result of this type.

It is well known that a graph group $G(\Sigma, I)$ can be embedded into a right-angled Coxeter group [20]. For this, one has to duplicate each letter from Σ. Formally, we can take the right-angled Coxeter group $C(\Gamma, I_\Gamma)$ (in which $\bar{a}$ does not denote the inverse of a). Consider the mapping $\varphi(a) = a\bar{a}$ from Γ to Γ^*. Obviously, φ induces a homomorphism from $G(\Sigma, I)$ to the Coxeter group $C(\Gamma, I_\Gamma)$. As $\mathrm{IRR}(S_G) \subseteq M(\Gamma, I_\Gamma)$ is mapped to $\mathrm{IRR}(S_C) \subseteq M(\Gamma, I_\Gamma)$, we recover the well-known fact that φ is injective. Actually we see more. Assume that $\widehat{w}$ is the shortlex normal form of some $\varphi(g)$. Then replacing in $\widehat{w}$ factors $a\bar{a}$ with a and replacing factors $\bar{a}a$ with $\bar{a}$ yields a logspace reduction of the problem of computing shortlex normal forms in graph groups to the problem of computing shortlex normal forms in right-angled Coxeter groups. Thus, for our purposes it is enough to calculate shortlex normal forms for right-angled Coxeter groups of type $C(\Sigma, I)$. For the latter, it suffices to compute in logspace on input $u \in \Sigma^*$ some trace (or word) v such that $u = v$ in $C(\Sigma, I)$ and $|v| = \|u\|$. Then, the shortlex normal form for u is the lexicographic normal form of the trace v, which can be easily computed in logspace from u.

A trace in $M(\Sigma, I)$ is called a *Coxeter-trace* , if it does not have any factor a^2 where $a \in \Sigma$. It follows that every element in $C(\Sigma, I)$ has a unique representation as a Coxeter-trace. Let $a \in \Sigma$. A trace u is called a-short, if during the derivation $u \stackrel{*}{\Longrightarrow}_{S_C} \widehat{u} \in \mathrm{IRR}(S_C)$ the rule $a^2 \longrightarrow 1$ is not applied. Thus, u is a-short if and only if the number of occurrences of the letter a is the same in the trace u and its Coxeter-trace $\widehat{u}$. We are interested in the set of letters which survive the reduction process. By $\widehat{\alpha}(u) = \alpha(\widehat{u})$ we denote the alphabet of the unique Coxeter-trace $\widehat{u}$ with $u = \widehat{u}$ in $C(\Sigma, I)$. Here is a crucial observation:

Lemma 2. *A trace u is a-short if and only if u has no factor ava such that $\widehat{\alpha}(v) \subseteq I(a)$.*

Proof. If u contains a factor ava such that $\widehat{\alpha}(v) \subseteq I(a)$, then u is clearly not a-short. We prove the other direction by induction on the length of u. Write $u = a_1 \cdots a_n$ with $a_i \in \Sigma$. We identify u with its dependence graph $\mathrm{DG}(u)$ which has vertex set $\{1, \ldots, n\}$. Assume that u is not a-short. Then, during the derivation $u \stackrel{*}{\Longrightarrow}_{S_C} \widehat{u}$, for a first time a vertex i with label $a_i = a$ is canceled with vertex j with label $a_j = a$ and $i < j$. It is enough to show that $\widehat{\alpha}(a_{i+1} \cdots a_{j-1}) \subseteq I(a)$. If the cancellation of i and j happens in the first step of the rewriting process, then we are done: $\alpha(a_{i+1} \cdots a_{j-1}) \subseteq I(a)$. So, let the first step cancel vertices k and ℓ with labels $a_k = a_\ell = b$ and $k < \ell$. Clearly, $\{i, j\} \cap \{k, \ell\} = \emptyset$. The set $\widehat{\alpha}(a_{i+1} \cdots a_{j-1})$ can change, only if either $i < k < j < \ell$ or $k < i < \ell < j$. However in both cases we must have $(b, a) \in I$, and we are done by induction. $\square$

The standard geometric representation $\sigma : C(\Sigma, I) \to \mathrm{GL}(n, \mathbb{Z})$ (where $n = |\Sigma|$) is defined as follows (see [3]), where we write σ_a for the mapping $\sigma(a)$:

$$\sigma_a(a) = -a, \quad \sigma_a(b) = b \text{ if } (a, b) \in I, \quad \sigma_a(b) = b + 2a \text{ if } (a, b) \in D \text{ and } a \neq b.$$

In this definition, a, b are letters. We identify $\mathbb{Z}^n = \mathbb{Z}^\Sigma$ and vectors from $\mathbb{Z}^n$ are written as formal sums $\sum_b \lambda_b b$. One can easily verify that $\sigma_{ab}(c) = \sigma_{ba}(c)$ for $(a, b) \in I$ and $\sigma_{aa}(b) = b$. Thus, σ defines indeed a homomorphism from $C(\Sigma, I)$ to $\mathrm{GL}(n, \mathbb{Z})$ (as well as homomorphisms from Σ^* and from $M(\Sigma, I)$ to $\mathrm{GL}(n, \mathbb{Z})$). Note that if $w = uv$ is a trace and $(b, v) \in I$ for a symbol b, then $\sigma_w(b) = \sigma_u(b)$. The following proposition is fundamental for understanding how the internal structure of w is reflected by letting σ_w act on letters (called *simple roots* in the literature). For lack of a reference for this variant (of a well-known general fact) and since the proof is rather easy in the right-angled case (in contrast to the general case), we give a proof. Our proof is purely combinatorial.

Proposition 1. *Let wd be a Coxeter-trace, $\sigma_w(d) = \sum_b \lambda_b b$ and $wd = udv$ where ud is prime and $(d, v) \in I$. Then it holds:*

(1) $\lambda_b \neq 0 \iff b \in \alpha(ud)$. Moreover, $\lambda_b > 0$ for all $b \in \alpha(ud)$.
(2) Let $b, c \in \alpha(ud)$, $b \neq c$, and assume that the first b in $\mathrm{DG}(ud)$ appears before the first c in $\mathrm{DG}(ud)$. Then we have $\lambda_b > \lambda_c > 0$.

Proof. We prove both statements of the lemma by induction on $|u|$. For $|u| = 0$ both statements are clear. Hence, let $u = au'$ and $\sigma_{u'}(d) = \sum_b \mu_b b$. Thus,

$$\sigma_u(d) = \sum_b \lambda_b b = \sigma_a(\sum_b \mu_b b) = \sum_b \mu_b \sigma_a(b).$$

Note that $\mu_b = \lambda_b$ for all $b \neq a$. Hence, by induction $\lambda_b = 0$ for all $b \notin \alpha(ud)$ and $\lambda_b > 0$ for all $b \in \alpha(ud) \setminus \{a\}$.

Let us now prove (2) for the trace u (it implies $\lambda_a > 0$ and hence (1)). Consider $b, c \in \alpha(ud)$, $b \neq c$, such that the first b in $\mathrm{DG}(ud)$ appears before the first c in $\mathrm{DG}(ud)$. Clearly, this implies $c \neq a$. For $b \neq a$ we obtain that the first b in $\mathrm{DG}(u'd)$ appears before the first c in $\mathrm{DG}(u'd)$. Hence, by induction we get $\mu_b > \mu_c > 0$. Claim (2) follows since $b \neq a \neq c$ implies $\mu_b = \lambda_b$ and $\mu_c = \lambda_c$.

Thus, let $a = b$. As there is path from the first a to every c in $\mathrm{DG}(ud)$ we may replace c by the first letter we meet on such a path. Hence we may assume that a and c are dependent. Note that $a \neq c$ because u is a Coxeter-trace. Hence, $\lambda_c = \mu_c > 0$ and it is enough to show $\lambda_a > \mu_c$. But $\lambda_a \geq 2\mu_c - \mu_a$ by the definition of σ_a. If $\mu_a = 0$, then $\lambda_a \geq 2\mu_c$, which implies $\lambda_a > \mu_c$, since $\mu_c > 0$. Thus, we may assume $\mu_a > 0$. By induction, we get $a \in \alpha(u'd)$. Here comes the crucial point: the first c in $\mathrm{DG}(u'd)$ must appear before the first a in $u'd$. Thus, $\mu_c > \mu_a$ by induction, which finally implies $\lambda_a \geq 2\mu_c - \mu_a = \mu_c + (\mu_c - \mu_a) > \mu_c$. □

Corollary 1. *Let $C(\Sigma, I)$ be a fixed right-angled Coxeter group. Then on input $w \in \Sigma^*$ we can calculate in logspace the alphabet $\widehat{\alpha}(w)$ of the corresponding Coxeter-trace $\widehat{w}$.*

Proof. Introduce a new letter x which depends on all other letters from Σ. We have $\sigma_w(x) = \sigma_{\widehat{w}}(x) = \sum_b \lambda_b b$. As $\widehat{w}x$ is a Coxeter-trace and prime, we have for all $b \in \Sigma$: $b \in \widehat{\alpha}(w) \iff b \in \alpha(\widehat{w}x) \iff \lambda_b \neq 0$, where the last equivalence follows from Prop. 1. Whether $\lambda_b \neq 0$ can be checked in logspace, by computing $\lambda_b \bmod m$ for all numbers $m \leq |w|$, since the least common multiple of the first n numbers is larger than 2^n (if $n \geq 7$) and the λ_b are integers with $|\lambda_b| \leq 2^{|w|}$. See also [22] for an analogous statement in the general context of linear groups. □

The hypothesis in Cor. 1 of being right-angled will be removed in Thm. 2. It remains open whether this hypothesis can be removed in the following theorem.

Theorem 1. *Let G be a fixed graph group or a fixed right-angled Coxeter group. Then we can calculate in logspace shortlex normal forms in G.*

Proof. As remarked earlier, it is enough to consider a right-angled Coxeter group $G = C(\Sigma, I)$. Fix a letter $a \in \Sigma$. We first construct a logspace transducer, which computes for an input trace $w \in M(\Sigma, I)$ a trace $u \in M(\Sigma, I)$ with the following properties: (i) $u = w$ in $C(\Sigma, I)$, (ii) u is a-short, and (iii) for all $b \in \Sigma$, if w is b-short, then also u is b-short. Having such a logspace transducer for every $a \in \Sigma$, we can compose all of them in an arbitrary order (note that $|\Sigma|$ is a constant) to obtain a logspace transducer which computes for a given input

trace $w \in M(\Sigma, I)$ a trace u such that $w = u$ in $C(\Sigma, I)$ and u is a-short for all $a \in \Sigma$, i.e., $u \in \mathrm{IRR}(S_C)$. Thus $u = \widehat{w}$. From u we can compute easily in logspace the Hasse diagram of $\mathrm{DG}(u)$ and then the shortlex normal form.

So, let us fix a letter $a \in \Sigma$ and an input trace $w = a_1 \cdots a_n$, where $a_1, \ldots, a_n \in \Sigma$. We remove from left to right positions (or vertices) labeled by the letter a which cancel and which therefore do not appear in $\widehat{w}$. We read $a_1 \cdots a_n$ from left to right. In the i-th stage do the following: If $a_i \neq a$ output the letter a_i and switch to the $(i+1)$-st stage. If however $a_i = a$, then compute in logspace (using Cor. 1) the maximal index $j > i$ (if it exists) such that $a_j = a$ and $\widehat{\alpha}(a_{i+1} \cdots a_{j-1}) \subseteq I(a)$. If no such index j exists, then append the letter a_i to the output tape and switch to the $(i+1)$-st stage. If j exists, then append the word $a_{i+1} \cdots a_{j-1}$ to the output tape, but omit all a's. After that switch immediately to stage $j+1$. Let w_{i-1} be the content of the output tape at the beginning of stage i (hence, $w_0 = 1$). The invariant of the algorithm is that (i) $w_{i-1} = a_1 \cdots a_{i-1}$ in $C(\Sigma, I)$, (ii) w_{i-1} is a-short, and (iii) if $a_1 \cdots a_{i-1}$ is b-short, then also w_{i-1} is b-short. The proof of this fact uses Lem. 2. □

5 Arbitrary Coxeter groups

In this section G denotes a fixed (not necessarily right-angled) *Coxeter group*, which is given by a generating set $\Sigma = \{a_1, \ldots, a_n\}$ of n generators and a symmetric $n \times n$ matrix $M = (m_{i,j})_{1 \leq i,j \leq n}$ over $(\mathbb{N} \setminus \{0\}) \cup \{\infty\}$ such that $m_{i,j} = 1 \iff i = j$. The defining relations are $(a_i a_j)^{m_{i,j}} = 1$ for $1 \leq i, j \leq n$ with $m_{i,j} < \infty$. In particular, $a_i^2 = 1$ for $1 \leq i \leq n$. One can show that if u and v are geodesics with $u = v$ in G then $\alpha(u) = \alpha(v)$ [3, Cor. 1.4.8] (Recall that $\alpha(x)$ denotes the alphabet of the word x). We will show how to compute this alphabet in logspace. We fix the standard geometric representation $\sigma : G \to \mathrm{GL}(n, \mathbb{R})$ (where we write again σ_w for the mapping $\sigma(w)$), see e.g. [3, Sect. 4.2]:

$$\sigma_{a_i}(a_j) = a_j + 2\cos(\pi/m_{i,j}) \cdot a_i$$

Let $\mathbb{R}^\Sigma$ be the n dimensional real vector space where the letter a_i is identified with the i-th unit vector. Thus, vectors can be written as formal sums $\sum_{b \in \Sigma} \lambda_b b$ with real coefficients λ_b. We write $\sum_{b \in \Sigma} \lambda_b b \geq 0$ if $\lambda_b \geq 0$ for all $b \in \Sigma$. The following lemma can be found in [3, Prop. 4.2.5]:

Lemma 3. *Let $w \in G$, $a \in \Sigma$. We have $\sigma_w(a) \geq 0$ if and only if $\|wa\| > \|w\|$.*

As in the proof of Cor. 1 introduce a new letter x with $x^2 = 1$, but no other new defining relation. This yields a Coxeter group $G' = G * (\mathbb{Z}/2\mathbb{Z}) \geq G$ generated by $\Sigma' = \Sigma \cup \{x\}$. Thus, ax is of infinite order in G' for all $a \in \Sigma$. Clearly, $\|wx\| > \|w\|$ for all $w \in G$. Hence, $\sigma_w(x) \geq 0$ for all $w \in G$ by Lem. 3.

Lemma 4. *Let $w \in G$ and $\sigma_w(x) = \sum_{b \in \Sigma'} \lambda_b b$. Then for all $b \in \Sigma$ we have $\lambda_b \neq 0$ if and only if the letter b appears in the shortlex normal form of w.*

Proof. We may assume that w is a geodesic in G. We prove the result by induction on $\|w\| = |w|$. If $w = 1$, then the assertion is trivial. If $b \in \Sigma$ does not occur as a letter in w, then it is clear that $\lambda_b = 0$. Thus, we may assume that $b \in \alpha(w)$ and we have to show that $\lambda_b \neq 0$. By induction, we may write $w = ua$ with $\|uax\| > \|ua\| > \|u\|$. We have $\sigma_w(x) = \sigma_u\sigma_a(x) = \sigma_u(x+2a) = \sigma_u(x)+2\sigma_u(a)$. The standard geometric representation yields moreover $\sigma_w(x) = x + \sum_{c\in\Sigma} \lambda_c c$, where $\lambda_c \geq 0$ for all $c \in \Sigma$ by Lem. 3. As $\|ua\| > \|u\|$ we get $\sigma_u(a) \geq 0$ by Lem. 3. Moreover, by induction (and the fact $\|ux\| > \|u\|$), we know that for all letters $c \in \alpha(u)$ the corresponding coefficient in $\sigma_u(x)$ is strictly positive. Thus, we are done if $b \in \alpha(u)$. So, the remaining case is that $b = a \notin \alpha(u)$. However, in this case $\sigma_u(a) = a + \sum_{c\in\Sigma\setminus\{a\}} \mu_c c$. Hence $\lambda_a \geq 2$. □

Theorem 2. *There is a logspace transducer which on input $w \in \Sigma^*$ computes the set of letters occurring in the shortlex normal form of w.*

Proof. Using the technique from [22] and Lem. 4, we can carry out all computations in the polynomial ring $\mathbb{Z}[X]$ [22]. In order to check that entries are not zero it suffices to check it $\bmod\ m$ with respect to all m up to a polynomial threshold. Due to space limitations, details are skipped. □

6 Free partially commutative inverse monoids

A monoid M is *inverse*, if for every $x \in M$ there is $\overline{x} \in M$ with

$$x\overline{x}x = x, \quad \overline{x}x\overline{x} = \overline{x}, \text{ and } \quad x\overline{x}\,y\overline{y} = y\overline{y}\,x\overline{x}. \tag{1}$$

The element $\overline{x}$ is uniquely defined by these properties and it is called the *inverse* of x. Thus, we may also use the notation $\overline{x} = x^{-1}$. It is easy to see that every idempotent element in an inverse monoid has the form xx^{-1}, and all these elements are idempotent. Using equations (1) for all $x, y \in \Gamma^*$ as defining relations we obtain the *free inverse monoid* $\mathrm{FIM}(\Sigma)$ which has been widely studied in the literature. More details on inverse monoids can be found in [21].

An *inverse monoid over an independence alphabet* (Σ, I) is an inverse monoid M together with a mapping $\varphi : \Sigma \to M$ such that $\varphi(a)\varphi(b) = \varphi(b)\varphi(a)$ and $\overline{\varphi(a)}\varphi(b) = \varphi(b)\overline{\varphi(a)}$ for all $(a, b) \in I$. We define the *free partially commutative inverse monoid over* (Σ, I) as the quotient monoid

$$\mathrm{FIM}(\Sigma, I) = \mathrm{FIM}(\Sigma)/\{ab = ba, \overline{a}b = b\overline{a} \mid (a, b) \in I\}.$$

It is an inverse monoid over (Σ, I). Da Costa has studied $\mathrm{FIM}(\Sigma, I)$ in his PhD thesis [31]. He proved that $\mathrm{FIM}(\Sigma, I)$ has a decidable word problem, but he did not show any complexity bound. The first upper complexity bound for the word problem is due to [12], where it was shown to be solvable in time $O(n \log(n))$ on a RAM. The aim of this section is to show that the space complexity of the word problem of $\mathrm{FIM}(\Sigma, I)$ is very low, too.

Theorem 3. *The word problem of* $\mathrm{FIM}(\Sigma, I)$ *can be solved in logspace.*

Proof. For a word $u = a_1 \cdots a_n$ ($a_1, \ldots, a_n \in \Gamma$) let $u_i \in M(\Gamma, I_\Gamma)$ ($1 \leq i \leq n$) be the trace represented by the prefix $a_1 \cdots a_i$ and define

$$M(u) = \{p \mid \exists 1 \leq i \leq n : p \text{ is a prime prefix of } \widehat{u_i}\} \subseteq M(\Gamma, I_\Gamma). \tag{2}$$

(This set is a partial commutative analogue of the classical notion of *Munn tree* introduced in [26].) It is shown in [12, Sect. 3] that for all words $u, v \in \Gamma^*$, $u = v$ in $\mathrm{FIM}(\Sigma, I)$ if and only if (i) $u = v$ in the graph group $G(\Sigma, I)$ and (ii) $M(u) = M(v)$. Since $G(\Sigma, I)$ is linear, condition (i) can be checked in logspace [22, 30]. For (ii), it suffices to show that the set $M(u)$ from (2) can be computed in logspace from the word u (then $M(u) = M(v)$ can be checked in logspace, since the word problem for the trace monoid $M(\Gamma, I_\Gamma)$ belongs to uniform TC^0 [1] and hence to logspace). By Thm. 1 we can compute in logspace a list of all normal forms $\widehat{u_i}$ ($1 \leq i \leq n$), where u_i is the prefix of u of length i. By composing this logspace transducer with a logspace transducer for computing prime prefixes (see Lem. 1), we obtain a logspace transducer for computing the set $M(u)$. □

7 Concluding remarks and open problems

We have shown that shortlex normal forms can be computed in logspace for graph groups and right-angled Coxeter groups. For general Coxeter groups, we are only able to compute the set of letters appearing in the shortlex normal form in logspace. An obvious open problem is, whether for every Coxeter group shortlex normal forms can be computed in logspace. We are tempted to believe that this is indeed the case. A more general question is, whether shortlex normal forms can be computed in logspace for automatic groups. Here, we are more sceptical. It is not known whether the word problem of an arbitrary automatic group can be solved in logspace. In [23], an automatic *monoid* with a P-complete word problem was constructed. In fact, it is even open, whether the word problem for a hyperbolic group belongs to logspace. The best current upper bound is LOGCFL [23]. So, one might first try to lower this bound e.g. to $\mathsf{LOGDCFL}$. M. Kapovich pointed out that there are non-linear hyperbolic groups. Hence the results of [22, 30] (linear groups have logspace word problems) do not help here.

References

1. C. Àlvarez and J. Gabarró. The parallel complexity of two problems on concurrency. *Inform. Process. Lett.*, 38:61–70, 1991.
2. M. Bestvina and N. Brady. Morse theory and finiteness properties of groups. *Invent. Math.*, 129:445–470, 1997.
3. A. Björner and F. Brenti. *Combinatorics of Coxeter groups.* Springer, 2005.
4. R. Book and F. Otto. *String-Rewriting Systems.* Springer-Verlag, 1993.
5. B. Brink and R. B. Howlett. A finiteness property and an automatic structure for Coxeter groups. *Math. Ann.*, 296:179–190, 1993.
6. J.-Y. Cai. Parallel computation over hyperbolic groups. In *Proceedings STOC'92*, 106–115. ACM Press, 1992.

7. W. A. Casselman. Automata to perform basic calculations in Coxeter groups. *C.M.S. Conference Proceedings*, 16, 1994.
8. J. Crisp, E. Godelle, and B. Wiest. The conjugacy problem in right-angled Artin groups and their subgroups. *J. Topol.*, 2(3), 2009.
9. J. Crisp and B. Wiest. Embeddings of graph braid and surface groups in right-angled artin groups and braid groups. *Algebr. Geom. Topol.*, 4:439–472, 2004.
10. M. W. Davis. *The geometry and topology of Coxeter groups*, volume 32 of *London Mathematical Society Monographs Series*. Princeton University Press, 2008.
11. V. Diekert. *Combinatorics on Traces*. LNCS 454. Springer, 1990.
12. V. Diekert, M. Lohrey, and A. Miller. Partially commutative inverse monoids. *Semigroup Forum*, 77:196–226, 2008.
13. V. Diekert and A. Muscholl. Solvability of equations in free partially commutative groups is decidable. *Internat. J. Algebra Comput.*, 16:1047–1070, 2006. Journal version of ICALP 2001, 543–554, LNCS 2076.
14. C. Droms, J. Lewin, and H. Servatius. The length of elements in free solvable groups. *Proc. Amer. Math. Soc.*, 119:27–33, 1993.
15. M. Elder. A linear-time algorithm to compute geodesics in solvable Baumslag-solitar groups. *Illinois J. Math.*, 54:109–128, 2010.
16. M. Elder, G. Elston, and G. Ostheimer. On groups that have normal forms computable in logspace. AMS Sectional Meeting, Las Vegas, May 2011. Paper in preparation.
17. M. Elder and A. Rechnitzer. Some geodesic problems in groups. *Groups. Complexity. Cryptology*, 2:223–229, 2010.
18. D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston. *Word Processing in Groups*. Jones and Bartlett, Boston, 1992.
19. R. Ghrist and V. Peterson. The geometry and topology of reconfiguration. *Adv. in Appl. Math.*, 38:302–323, 2007.
20. T. Hsu and D. T. Wise. On linear and residual properties of graph products. *Michigan Mathematical Journal*, 46(2):251–259, 1999.
21. M. V. Lawson. *Inverse Semigroups: The Theory of Partial Symmetries*. World Scientific, 1999.
22. R. J. Lipton and Y. Zalcstein. Word problems solvable in logspace. *J. Assoc. Comput. Mach.*, 24:522–526, 1977.
23. M. Lohrey. Decidability and complexity in automatic monoids. *Internat. J. Found. Comput. Sci.*, 16:707–722, 2005.
24. M. Lohrey and N. Ondrusch. Inverse monoids: Decidability and complexity of algebraic questions. *Inf. Comput.*, 205:1212–1234, 2007.
25. C. F. Miller III. Decision problems for groups – survey and reflections. In *Algorithms and Classification in Combinatorial Group Theory*, 1–60. Springer, 1992.
26. W. Munn. Free inverse semigroups. *Proc. London Math. Soc.*, 29:385-404, 1974.
27. A. Myasnikov, V. Roman'kov, A. Ushakov, and A.Vershik. The word and geodesic problems in free solvable groups. *Trans. Amer. Math. Soc.*, 362:4655–4682, 2010.
28. Ch. Papadimitriou. *Computation Complexity*. Addison-Wesley, 1994.
29. M. Paterson and A. Razborov. The set of minimal braids is co-NP-complete. *J. Algorithms*, 12:393–408, 1991.
30. H.-U. Simon. Word problems for groups and contextfree recognition. In *Proceedings FCT'79*, 417–422. Akademie-Verlag, 1979.
31. A. A. Veloso da Costa. *Γ-Produtos de Monóides e Semigrupos*. PhD thesis, Universidade do Porto, Faculdade de Ciências, 2003.
32. S. Waack. Tape complexity of word problems. In *Proceedings FCT'81*, LNCS 117, 467–471. Springer, 1981.