Logspace computations in Coxeter groups and graph groups

Volker Diekert, Jonathan Kausch, and Markus Lohrey

ABSTRACT. Computing normal forms in groups (or monoids) is computationally harder than solving the word problem (equality testing), in general. 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.

For Coxeter groups we show that the following algorithmic tasks can be solved by a deterministic Turing machine using logarithmic work space, only: 1. Compute the length of any geodesic normal form. 2. Compute the set of letters occurring in any geodesic normal form. 3. Compute the Parikh-image of any geodesic normal form in case that all defining relations have even length (i.e., in even Coxeter groups.) 4. For right-angled Coxeter groups we can actually compute the shortlex normal form in logspace.

Next, we apply the results to right-angled Artin groups. They are also known as free partially commutative groups or as graph groups. As a consequence of our result on right-angled Coxeter groups we show that shortlex normal forms in graph groups can be computed in logspace, too. Graph groups play an important rôle in group theory, and they have a close connection to concurrency theory. 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. Concurrent systems which are deterministic and co-deterministic can be studied via inverse monoids.

1. Introduction

The study of group theoretical decision problems, like the word problem: "Is a given word equal to 1 in the group?", the conjugacy problem: "Are two given words conjugate 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 (e.g. Tietze 1908, Dehn 1911). More details are in the survey [33].

With the emergence of computational complexity theory, the complexity of these decision problems in various classes of groups has developed into an active 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 theoretical problems which can be solved efficiently in parallel. More precisely, we are interested in *deterministic*

logspace, called simply *logspace* in the following. The class logspace is at a lower level in the NC-hierarchy[1] of parallel complexity classes:

$$\mathsf{NC}^1 \subseteq \mathsf{LOGSPACE} \subseteq \mathsf{NC}^2 \subseteq \mathsf{NC}^3 \subseteq \cdots \subseteq \mathsf{NC} = \bigcup_{i \geq 1} \mathsf{NC}^i \subseteq \mathsf{P}$$

For more details about circuit complexity we refer to [**40**]. It is a standard conjecture in complexity theory that NC and P are different, just like a standard conjecture says $\mathsf{P} \neq \mathsf{NP}$. But this is debatable, of course.

A fundamental result in this context was shown in [**29, 38**]: The word problem of finitely generated linear groups belongs to logspace. In [**29**], Lipton and Zalcstein proved this result for fields of characteristic 0. The case of a field of prime characteristic was considered in [**38**] by Simon. The class of groups with a word problem in logspace is further investigated in [**41**]. Another important result is Cai's NC^2 algorithm for the word problem of a hyperbolic group [**6**]. In [**31**] this result was improved to LOGCFL, which is the class of all languages that are logspace-reducible to a context-free language. LOGCFL is a subclass of NC^2 and hence in the intersection of the class of problems which can be decided in polynomial time and the class of problems which can be decided in space $\log^2(n)$. As a parallel complexity class LOGCFL coincides with the (uniform) class SAC^1.

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 [**18, 19, 21, 27, 35, 37**]. 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 [**18**], 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 [**35**] 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* [**22**] (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. Examples of shortlex automatic groups are Coxeter groups, Artin groups of large type, and hyperbolic groups. So for all these classes, shortlex normal forms can be computed in quadratic time. In [**35**], it is also noted that geodesics in nilpotent groups (which are in general not automatic) 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

---

[1]The notation NC refers to Nicholas "Nick" Pippinger. Nick's class NC^i denotes the class of languages that can be accepted by (uniform) Boolean circuits of polynomial size and polylogarithmic depth $O(\log^i(n))$, where all gates have fan-in ≤ 2.

$\mathcal{O}(\log n)$, 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 machine. Still better, it can be simulated 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 [36].

Recently, the class of groups, where geodesics and shortlex normal forms can be computed in logspace, attracted attention, see [20], 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 [32].) In this paper, we deal with the problem of computing shortlex normal forms for Coxeter groups. Coxeter groups are discrete reflection groups and play an important role in many parts of mathematics, see [3, 12]. Every Coxeter group is linear and therefore has a logspace word problem. Moreover, as mentioned above, Coxeter groups are shortlex automatic [5, 7]. Therefore shortlex normal forms can be computed in quadratic time. However, no efficient parallel algorithms are known so far. In particular, it is open whether shortlex normal forms in Coxeter groups can be computed in logspace. We do not solve this problem in this paper, but we are able to compute in logspace some important invariants of geodesics. More precisely, we are able to compute in logspace (i) the length of the shortlex normal form of a given element (Theorem 3.7) and (ii) the alphabet of symbols that appear in the shortlex normal form of a given element (Theorem 3.9). The proofs for these results combine non-trivial results for Coxeter groups with some advanced tools from computational algebra. More precisely, we use the following results:

- The Chinese remainder representation of a given number m (which is the tuple of remainders $m \mod p_i$ for the first k primes $p_1, \ldots, p_k$, where $m < p_1 p_2 \cdots p_k$) can be transformed in logspace into the binary representation of m [8, 24]. This result is the key for proving that iterated multiplication and division can be computed in logspace.
- Arbitrary algebraic constants can be approximated in logspace up to polynomially many bits. This result was recently shown in [11, 26].

For the case of even Coxeter groups, i.e., Coxeter groups where all defining relations have even length, we can combine Theorem 3.7 and Theorem 3.9 in one more general result, saying that the Parikh-image of the shortlex normal form can be computed in logspace (Theorem 3.12). The Parikh-image of a word $w \in \Sigma^*$ is the image of w under the canonical homomorphism from Σ^* to $\mathbb{N}^{|\Sigma|}$.

As mentioned above, it remains open, whether shortlex normal forms in Coxeter groups can be computed in logspace. In the second part of this paper, we prove that for the important subclass of *right-angled Coxeter groups* shortlex normal forms can be computed in logspace (Theorem 5.4). A right-angled Coxeter group is defined by a finite undirected graph (Σ, I) by taking Σ as the set of group generators and adding the defining relations $a^2 = 1$ for all $a \in \Sigma$ and $ab = ba$ for all edges $(a, b) \in I$. We use techniques from the theory of Mazurkiewicz traces [13]. 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 provide an elementary proof that the alphabet of symbols that

appear in a geodesic for g can be computed in logspace from g (Corollary 5.3).[2] In contrast to general Coxeter groups, for right-angled Coxeter groups this alphabetic information suffices in order to compute shortlex normal forms in logspace.

Right-angled Coxeter groups are closely related to graph groups, which 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 relations $ab = ba$ for all edges $(a, b) \in I$. Hence, a right-angled Coxeter group is obtained from a graph group by adding all relations $a^2 = 1$ for all generators a. Graph groups received in recent years a lot of attention in group theory because of their rich subgroup structure [**2, 10, 23**]. On the algorithmic side, (un)decidability results were obtained for many important group-theoretic decision problems in graph groups [**9, 16**]. There is a standard embedding of a graph group into a right-angled Coxeter group [**25**]. Hence, also graph groups are linear and have logspace word problems. Using the special properties of this embedding, we can show that also for graph groups, shortlex normal forms can be computed in logspace (Theorem 5.4). We remark that this is an optimal result in the sense that logspace is the smallest known complexity class for the word problem in free groups already. Clearly, computing shortlex normal forms is at least as difficult than solving the word problem.

Finally, we apply Theorem 5.4 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 [**15**] 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 [**39**].) 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 (Theorem 6.1). 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 [**32**]. Let us emphasize that in order to obtain Theorem 6.1 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).

The work on this paper started at the AMS Sectional Meeting, Las Vegas, May 2011, and was motivated by the lecture of Gretchen Ostheimer [**20**]. A preliminary version of our results appeared as a conference abstract at the Latin American Symposium on Theoretical Informatics (LATIN 2012), [**14**]. In contrast to the conference abstract this paper provides full proofs and it contains new material about even Coxeter groups and how to compute geodesic lengths in all Coxeter groups.

---

[2]In contrast, the proof of Theorem 3.9, which generalizes Corollary 5.3 to all Coxeter groups, is more difficult in the sense that it uses geometry and more facts from [**3**].

2. Notation

Throughout Σ (resp. Γ) denotes a finite *alphabet.* This is a finite set, sometimes equipped with a linear order. An element of Σ is called a *letter.* By Σ^* we denote the free monoid over Σ. For a word $w \in \Sigma^*$ we denote by $\alpha(w)$ the *alphabet of w*: it is the set of letters occurring in w. With $|w|$ we denote the length of w. The *empty word* has length 0; and it is denoted by 1 as other neutral elements in monoids or groups.

All groups and monoids M in this paper are assumed to be finitely generated; and they come with a monoid homomorphism $\pi : \Sigma^* \to M$. Frequently we assume that M comes with an involution[3] $x \mapsto \bar{x}$ on M, and then we require that $\pi(\Sigma) \cup \pi(\overline{\Sigma})$ generates M as a monoid.

If the monoid M is a group G, then the involution is always given by taking inverses, thus $\bar{x} = x^{-1}$. Moreover, G becomes a factor group of the *free group* $F(\Sigma)$ thanks to $\pi : \Sigma^* \to G$.

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: We let $\bar{\bar{a}} = a$ and $\overline{a_1 \cdots a_n} = \overline{a_n} \cdots \overline{a_1}$. Hence, we can lift our homomorphism $\pi : \Sigma^* \to M$ to a surjective monoid homomorphism $\pi : \Gamma^* \to M$ which respects the involution, i.e., $\pi(\bar{x}) = x^{-1}$.

Given a surjective homomorphism $\pi : \Gamma^* \to M$ and a linear order on Γ we can define the geodesic length and the shortlex normal form for elements in M as follows. For $w \in M$, the *geodesic length* $\|w\|$ is the length of a shortest word in $\pi^{-1}(w)$. The *shortlex normal form* of w is the lexicographical first word in the finite set $\left\{ u \in \pi^{-1}(w) \mid |u| = \|w\| \right\}$. By a *geodesic* we mean any word in the finite set $\left\{ u \in \pi^{-1}(w) \mid |u| = \|w\| \right\}$.

3. Coxeter groups

A *Coxeter group* G is given by a generating set $\Sigma = \{a_1, \ldots, a_n\}$ of n generators and a symmetric $n \times n$ matrix $M = (m_{i,j})_{1 \le i,j \le n}$ over $\mathbb{N}$ such that $m_{i,j} = 1$ if and only if $i = j$. The defining relations are $(a_i a_j)^{m_{i,j}} = 1$ for $1 \le i, j \le n$. In particular, $a_i^2 = 1$ for $1 \le i \le n$. Traditionally, one writes the entry ∞ instead 0 in the Coxeter matrix M and then $m_{i,j}$ becomes the order of the element $a_i a_j$.

A Coxeter group is called *even*, if all $m_{i,j}$ are even numbers for $i \ne j$. It is called *right-angled*, if $m_{i,j} \in \{0, 1, 2\}$ for all i, j. The defining relations of a right-angled Coxeter group can be rewritten in the following form: $a_i^2 = 1$ for $1 \le i \le n$ and $a_i a_j = a_j a_i$ for $(i, j) \in I$ where I denotes a symmetric and irreflexive relation $I \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$. Thus, one could say that a right-angled Coxeter group is a *free partially commutative* Coxeter group. Readers interested only in right-angled Coxeter groups or in the application to graph groups (i.e., right-angled Artin groups) may proceed directly to Section 4.

3.1. Computing the geodesic alphabet and geodesic length. Throughout this subsection G denotes a Coxeter group given by a fixed $n \times n$ matrix M as above. Recall that $\alpha(x)$ denotes the alphabet of the word x. 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]. We will

---

[3]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 $\overline{xy} = \bar{y}\,\bar{x}$.

show how to compute this alphabet in logspace. Moreover, we will show that also the geodesic length $|w|$ for a given $w \in G$ can be computed in logspace.

Let $\mathbb{R}^\Sigma$ be the n dimensional real vector space where the letter a is identified with the a-th unit vector. Thus, vectors will be written as formal sums $\sum_{b \in \Sigma} \lambda_b b$ with real coefficients $\lambda_b \in \mathbb{R}$. We fix the standard geometric representation $\sigma : G \to \mathrm{GL}(n, \mathbb{R})$, where we write σ_w for the mapping $\sigma(w)$, see e.g. [**3**, Sect. 4.2]:

$$(3.1) \qquad \sigma_{a_i}(a_j) = \begin{cases} a_j + 2\cos(\pi/m_{i,j}) \cdot a_i & \text{if } m_{i,j} \neq 0 \\ a_j + 2 \cdot a_i & \text{if } m_{i,j} = 0 \end{cases}$$

Note that for $a \in \Sigma$, $\sigma_w(a)$ cannot be the zero vector, since σ_w is invertible. We write $\sum_{b \in \Sigma} \lambda_b b \geq 0$ if $\lambda_b \geq 0$ for all $b \in \Sigma$. The following fundamental lemma can be found in [**3**, Prop. 4.2.5]:

LEMMA 3.1. *Let $w \in G$ and $a \in \Sigma$. We have*

$$\|wa\| = \begin{cases} \|w\| + 1 & \text{if } \sigma_w(a) \geq 0 \\ \|w\| - 1 & \text{if } \sigma_w(a) \leq 0 \end{cases}$$

LEMMA 3.2. *For a given $w \in G$ and $a, b \in \Sigma$, one can check in logspace, whether $\lambda_b \geq 0$, where $\sum_{b \in \Sigma} \lambda_b b = \sigma_w(a)$.*

In order to prove Lemma 3.2, we need several tools. Let p_i denote the ith prime number. It is well-known from number theory that the ith prime requires $O(\log(i))$ bits in its binary representation. For a number $0 \leq M < \prod_{i=1}^m p_i$ we define the *Chinese remainder representation* $\mathsf{CRR}_m(M)$ as the m-tuple

$$\mathsf{CRR}_m(M) = (M \bmod p_i)_{1 \leq i \leq m}.$$

By the Chinese remainder theorem, the mapping $M \mapsto \mathsf{CRR}_m(M)$ is a bijection from the interval $[0, \prod_{i=1}^m p_i - 1]$ to $\prod_{i=1}^m [0, p_i - 1]$. By the following theorem, we can transform a CRR-representation very efficiently into binary representation.

THEOREM 3.3 ([**8**, Thm. 3.3]). *For a given tuple $(r_1, \ldots, r_m) \in \prod_{i=1}^m [0, p_i - 1]$, we can compute in logspace the binary representation of the unique number $M \in [0, \prod_{i=1}^m p_i - 1]$ such that $\mathsf{CRR}_m(M) = (r_1, \ldots, r_m)$.*

By [**24**], the transformation from the CRR-representation to the binary representation can be even computed by $\mathsf{DLOGTIME}$-uniform TC^0-circuits. Our second tool is a gap theorem for values $p(\zeta)$, where $p(x) \in \mathbb{Z}[x]$ and ζ is a root of unity. For a polynomial $p(x) = \sum_{i=0}^n a_i x^i$ with integer coefficients a_i let $|p(x)| = \sum_{i=0}^n |a_i|$.

THEOREM 3.4 ([**30**, Thm. 3]). *Let $p(x) \in \mathbb{Z}[x]$ and let ζ be a dth root of unity such that $p(\zeta) \neq 0$. Then $|p(\zeta)| > |p(x)|^{-d}$.*

Finally, our third tool for the proof of Lemma 3.2 is the following result, which was recently shown (independently) in [**11**, **26**].

THEOREM 3.5 ([**11**, Thm .2],[**26**, Cor. 4.6]). *For every fixed algebraic number $\alpha \in \mathbb{R}$ the following problem can be computed in logspace:*
INPUT: A unary coded number n.
OUTPUT: The binary representation of the integer $\lfloor 2^n \alpha \rfloor$.

REMARK 3.6. The result of [**26**] is actually stronger showing that the output in Theorem 3.5 can be computed in uniform TC^0.

Proof of Lemma 3.2. We decompose the logspace algorithm into several logspace computations. The linear mapping σ_w can be written as a product of matrices $A_1 A_2 \cdots A_{|w|}$, where every A_i is an $(n \times n)$-matrix with entries from $\{0, 1, 2\} \cup \{2 \cos(\pi/m_{i,j}) \mid m_{i,j} \neq 0\}$ (which is the set of coefficients that appear in (3.1)). Then, we have to check whether this matrix product applied to the unit vector a has a non-negative value in the b-coordinate. This value is the entry $(A_1 A_2 \cdots A_{|w|})_{a,b}$ of the product matrix $A_1 A_2 \cdots A_{|w|}$.

Let m be the least common multiple of all $m_{i,j} \neq 0$; this is still a constant. Let $\zeta = e^{\pi i/m}$, which is a primitive $(2m)^{th}$ root of unity. If $m = m_{i,j} \cdot k$, we have

$$2 \cdot \cos\left(\frac{\pi}{m_{i,j}}\right) = \zeta^k + \zeta^{2m-k}.$$

Hence, we can assume that every A_i is an $(n \times n)$-matrix over $\mathbb{Z}[\zeta]$. We now replace ζ by a variable X in all matrices $A_1, \ldots, A_{|w|}$; let us denote the resulting matrices over the ring $\mathbb{Z}[X]$ with $B_1, \ldots, B_{|w|}$. Each entry in one of these matrices is a polynomial of degree $< 2m$ with coefficients bounded by 2. More precisely, for every entry $p(X)$ of a matrix B_i we have $|p(X)| \leq 2$. Let $|B_i|$ be the sum of all $|p(X)|$ taken over all entries of the matrix B_i. Hence, $|B_i| \leq 2n^2$.

Step 1. In a first step, we show that the product $B_1 \cdots B_{|w|}$ can be computed in logspace in the ring $\mathbb{Z}[X]/(X^{2m} - 1)$ (keeping in mind that $\zeta^{2m} = 1$). Every entry in the product $B_1 \cdots B_{|w|}$ is a polynomial of degree $< 2m$ with coefficients bounded in absolute value by $|B_1| \cdots |B_{|w|}| \leq (2n^2)^{|w|}$. Here n is a fixed constant. Hence, every coefficient in the matrix $B_1 \cdots B_{|w|}$ can be represented with $O(|w|)$ bits. In logspace, one can compute a list of the first k prime numbers $p_1, p_2, \ldots, p_k$, where $k \in O(|w|)$ is chosen such that $\prod_{i=1}^{k} p_i > (2n^2)^{|w|}$ [8]. Each p_i is bounded by $|w|^{O(1)}$.

For every $1 \leq i \leq k$, we can compute in logspace the matrix product $B_1 \cdots B_{|w|}$ in $\mathbb{F}_{p_i}[X]/(X^{2m}-1)$, i.e., we compute the coefficient of each polynomial in $B_1 \cdots B_{|w|}$ modulo p_i. In the language of [8]: For each coefficient of a polynomial in $B_1 \cdots B_{|w|}$, we compute its Chines remainder representation. From this representation, we can compute in logspace by Theorem 3.3 the binary representation of the coefficient. This shows that the product $B = B_1 \cdots B_{|w|}$ can be computed in the ring $\mathbb{Z}[X]/(X^{2m} - 1)$.

Step 2. We know that if X is substitued by ζ in the matrix B, then we obtain the product $A = A_1 \cdots A_{|w|}$ (the matrix we are actully interested in), which is a matrix over $\mathbb{R}$. Every entry of the matrix A is of the form $\sum_{j=0}^{2m-1} a_j \zeta^j$, where a_j is a number with $O(|w|)$ bits that we have computed in Step 1. If $\sum_{j=0}^{2m-1} a_j \zeta^j \neq 0$, then by Theorem 3.4, we have

$$\left| \sum_{j=0}^{2m-1} a_j \zeta^j \right| > \left(\sum_{j=0}^{2m-1} |a_j| \right)^{-2m}.$$

Since m is a constant, and $|a_j| \leq 2^{O(|w|)}$, we have

$$\sum_{j=0}^{2m-1} a_j \zeta^j = 0 \quad \text{or} \quad \left| \sum_{j=0}^{2m-1} a_j \zeta^j \right| > 2^{-c|w|}$$

for a constant c. Therefore, to check whether $\sum_{j=0}^{2m-1} a_j \zeta^j \geq 0$ or $\sum_{j=0}^{2m-1} a_j \zeta^j \leq 0$, it suffices to approximate this sum up to $c|w|$ many fractional bits. This is the goal of the second step.

Since we are sure that $\sum_{j=0}^{2m-1} a_j \zeta^j \in \mathbb{R}$, we can replace the sum symbolically by its real part, which is $\sum_{j=0}^{2m-1} a_j \cos(j\pi/m)$. In order to approximate this sum up to $c|w|$ many fractional bits, it suffices to approximate each $\cos(j\pi/m)$ up to $d|w|$ many fractional bits (recall that $a_j \in 2^{O(|w|)}$), where the constant d is large enough.

Every number $\cos(q \cdot \pi)$ for $q \in \mathbb{Q}$ is algebraic.[4] Theorefore, by Theorem 3.5, every number $\cos(j\pi/m)$ $(0 \leq j \leq 2m - 1)$ can be approximated in logspace up to $d|w|$ many fractional bits. This concludes the proof. $\square$

Lemma 3.2 can be used in order to compute in logspace the geodesic length $\|w\|$ for a given group element $w \in G$:

THEOREM 3.7. *For a given word $w \in \Sigma^*$, the geodesic length $\|w\|$ can be computed in logspace.*

PROOF. By Lemma 3.1, the following algorithm correctly computes $\|w\|$ for $w = a_1 \cdots a_k$.

$\ell := 0$;
for $i = 1$ **to** k **do**
    if $\sigma_{a_1 \cdots a_{i-1}}(a_i) \geq 0$ **then**
        $\ell := \ell + 1$
    else
        $\ell := \ell - 1$
    endif
endfor
return ℓ

By Lemma 3.2 it can be implemented in logspace. $\square$

We finally apply Lemma 3.2 in order to compute in logspace the set of all letters that occur in a geodesic for a given group element $w \in G$. As remarked before, this alphabet is independent of the particular geodesic for w.

Introduce a new letter $x \notin \Sigma$ with $x^2 = 1$, but no other new defining relation. This yields the Coxeter group $G' = G * (\mathbb{Z}/2\mathbb{Z})$ 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 Lemma 3.1.

LEMMA 3.8. *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 $\sigma_w(x) = \sigma_u \sigma_a(x) = \sigma_u(x + 2a) = \sigma_u(x) + 2\sigma_u(a)$.

---

[4]This seems to be a folklore fact, for which we could not find a reference. If $q \cdot \pi = \frac{2a\pi}{b}$ for $a \in \mathbb{Z}, b \in \mathbb{N}$, then by DeMoivre's formula $((\cos\theta + i\sin\theta)^n = \cos(n\theta) + i\sin(n\theta))$ we have $(\cos(q \cdot \pi) + i\sin(q \cdot \pi))^b = \cos(2a\pi) + i\sin(2a\pi) = 1$. Hence, $z := \cos(q \cdot \pi) + i\sin(q \cdot \pi)$ is algebraic, which implies that $\cos(q \cdot \pi) = \frac{1}{2}(z + \overline{z})$ is algebraic too.

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 Lemma 3.1. As $\|ua\| > \|u\|$ we get $\sigma_u(a) \geq 0$ by Lemma 3.1. 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 $\lambda_a \geq 2$. $\qquad\square$

THEOREM 3.9. *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. By Lemma 3.8, we have to check for every letter $b \in \Sigma$, whether $\lambda_b = 0$, where $\sum_{b \in \Sigma'} \lambda_b b = \sigma_w(x)$. By Lemma 3.2 (applied to the Coxeter group G') this is possible in logspace. $\qquad\square$

Let us remark that the use of Lemma 3.2 in the proof of Theorem 3.9 can be avoided, using the technique from [29] and Lemma 3.8. Every λ_b belongs to the ring $\mathbb{Z}[\zeta] \cong \mathbb{Z}[X]/\Phi(X)$, where ζ is a primitive $(2m)^{th}$ root of unity, $\Phi(X)$ is the $(2m)^{th}$ cyclotomic polynomial, and m is the least common multiple of all $m_{i,j} \neq 0$. In order to check whether $\lambda_b = 0$, we can check whether the value is zero mod r with respect to all r up to a polynomial threshold.

3.2. Computing the geodesic Parikh-image in even Coxeter groups. In this section we assume that G is an even Coxeter group. Thus, the entries $m_{i,j}$ of the matrix M are even for all $i \neq j$.

Let $a \in \Sigma$ be a letter and $w \in \Sigma^*$. By $|w|_a$ we denote the number of a's in a word $w \in \Gamma^*$. The *Parikh-image* of w is the vector $[\,|w|_a\,]_{a \in \Sigma} \in \mathbb{N}^\Sigma$. In other words, the Parikh-image of w is the image of w under the canonical homomorphism from the free monoid Σ^* to the free commutative monoid $\mathbb{N}^\Sigma$.

We show that for even Coxeter groups, the Parikh-image of geodesics can be computed in logspace. Actually, all geodesics for a given group element of an even Coxeter group have the same Parikh-image:

LEMMA 3.10. *Let G be an even Coxeter group and let $u, v \in \Sigma^*$ be geodesics with $u = v$ in G. Then we have $|u|_a = |v|_a$ for all $a \in \Sigma$.*

PROOF. Let $1 \leq i, j \leq n$ with $i \neq j$. Since the entry $m_{i,j}$ of the matrix M is even and $(a_i a_j)^{m_{i,j}} = 1$ in G, we get the relation $(a_i a_j)^{m_{i,j}/2} = (a_j a_i)^{m_{i,j}/2}$ which does not effect the Parikh-image. Let the string rewriting system T consist of all rules of the form $(a_i a_j)^{m_{i,j}/2} \to (a_j a_i)^{m_{i,j}/2}$ for $1 \leq i, j \leq n$ with $i \neq j$. Now, it follows from a well-known result of Tits (c.f. [3]) that if u and v are geodesic, then u can be transformed into v using applications of rules from T. Consequently, $|u|_a = |v|_a$ for all $a \in \Sigma$. $\qquad\square$

LEMMA 3.11. *Let G be an even Coxeter group, $a \in \Sigma$, and let u, w be geodesics such that $wa = u$ in G. Then there exists $\varepsilon \in \{1, -1\}$ such that $|u| = |w| + \varepsilon$ and $|u|_a = |w|_a + \varepsilon$. For all $b \in \Sigma \setminus \{a\}$ we have $|u|_b = |w|_b$.*

PROOF. By Lemma 3.1 there exists $\varepsilon \in \{1, -1\}$ with $|u| = |w| + \varepsilon$. Moreover, since $a^2 = 1$, we have $ua = w$ and $wa = u$ in G. Hence, if $|w| = |u| + 1$ (resp., $|u| = |w| + 1$), then ua and w (resp., wa and u) are geodesics defining the same group element in G. Lemma 3.10 implies that $|ua|_c = |w|_c$ (resp., $|wa|_c = |u|_c$) for all $c \in \Sigma$. This implies the conclusion of the lemma. $\qquad\square$

THEOREM 3.12. *Let G be an even Coxeter group. For a given word $w \in \Sigma^*$, the Parikh-image of the shortlex normal form for w can be computed in logspace.*

PROOF. Lemma 3.11 shows that the following straightforward modification of the logspace algorithm in (the proof of) Theorem 3.7 computes the Parikh-image of the shortlex normal form for w correctly. Let $w = a_1 \cdots a_k$ be the input word.

for all $a \in \Gamma$ **do**
    $\ell_a := 0$
endfor
for $i = 1$ **to** k **do**
    if $\sigma_{a_1 \cdots a_{i-1}}(a_i) \geq 0$ **then**
        $\ell_{a_i} := \ell_{a_i} + 1$
    else
        $\ell_{a_i} := \ell_{a_i} - 1$
    endif
endfor
return $[\ell_a]_{a \in \Gamma}$ $\square$

4. Mazurkiewicz traces and graph groups

In the rest of the paper, we will deal with right-angled Coxeter groups. As explained in Section 3, a right-angled Coxeter group can be specified by a finite undirected graph (Σ, I). The set Σ is the generating set and the relations are $a^2 = 1$ for all $a \in \Sigma$ and $ab = ba$ for all $(a, b) \in I$. Hence, I specifies a partial commutation relation, and elements of a right-angled Coxeter group can be represented by partially commutative words, also known as Mazurkiewicz traces. In this section, we will introduce some basic notions from the theory of Mazurkiewicz traces, see [**13, 17**] for more details.

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 relation with $I \subseteq I_\Gamma$ and such that $(a, b) \in I_\Gamma$ implies $(a, \overline{b}) \in I_\Gamma$. The independence alphabet (Σ, I) defines a *free partially commutative monoid* (or *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 4.1. *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 by examining all sequences $1 \leq i_1 < i_2 < \cdots < i_k = i$ with $k \leq |\Sigma|$. Such a sequence can be stored in logarithmic space since $|\Sigma|$ is a constant. $\square$

We use standard notations 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 4.2. The system $S_G = \{a\bar{a} \longrightarrow 1 \mid a \in \Gamma\}$ is (strongly) confluent and length-reducing over $M(\Gamma, I_\Gamma)$ [**13**]. The quotient monoid $M(\Gamma, I_\Gamma)/S_G$ is the graph group $G(\Sigma, I)$.

By Example 4.2 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 $\alpha(v) \subseteq I(a)$.

5. Right-angled Coxeter groups

Some of the results on right-angled Coxeter groups in this section are covered by more general statements in Section 3. However, the former section used quite involved tools from computational algebra and an advanced theory of Coxeter groups. In contrast the results we prove here on right-angled Coxeter groups are

purely combinatorial. Hence we can give simple and elementary proofs which makes this section fully self-contained. Moreover, in contrast to the general case, for the right-angled case we will succeed in computing shortlex normal forms in logspace.

Recall that a *right-angled Coxeter group* is specified by a finite undirected graph (Σ, I), i.e., an independence alphabet. The set Σ is the generating set and the relations are $a^2 = 1$ for all $a \in \Sigma$ and $ab = ba$ for all $(a, b) \in I$. We denote this right-angled Coxeter group by $C(\Sigma, I)$. 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 [**13**] 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 [**25**]. 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 $C(\Sigma, I)$ in logspace. 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 5.1. *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 $\alpha(a_{i+1} \cdots a_{j-1}) \subseteq I(a)$ and we are done. 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. $\qquad\square$

In the right-angled case, the standard geometric representation (see (3.1)) $\sigma : C(\Sigma, I) \to \mathrm{GL}(n, \mathbb{Z})$ (where $n = |\Sigma|$) can be defined as follows, where again we write σ_a for the mapping $\sigma(a)$:

$$
\begin{aligned}
\sigma_a(a) &= -a, \\
\sigma_a(b) &= \begin{cases} b & \text{if } (a, b) \in I, \\ b + 2a & \text{if } (a, b) \in D \text{ and } a \neq b. \end{cases}
\end{aligned}
$$

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, which is purely combinatorial.

PROPOSITION 5.2. *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) $\lambda_b \neq 0$ *if and only if $b \in \alpha(ud)$. 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,

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

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 $\lambda_a \geq 2\mu_c - \mu_a = \mu_c + (\mu_c - \mu_a) > \mu_c$. $\qquad \square$

COROLLARY 5.3. *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 $b \in \Sigma$:

$$b \in \widehat{\alpha}(w) \Longleftrightarrow b \in \alpha(\widehat{w}x) \Longleftrightarrow \lambda_b \neq 0,$$

where the last equivalence follows from Proposition 5.2. 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 [29] for an analogous statement in the general context of linear groups. $\qquad \square$

The hypothesis in Corollary 5.3 being a right-angled Coxeter group is actually not necessary as we have seen in Theorem 3.9. It remains open whether this hypothesis can be removed in the following theorem.

THEOREM 5.4. *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 Corollary 5.3) 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 $j + 1$. Here is a pseudo code description of the algorithm, where $\pi_{\Sigma \setminus \{a\}} : \Sigma^* \to (\Sigma \setminus \{a\})^*$ denotes the homomorphism that deletes all occurrences of a.

$i := 1;$
$w := 1;$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ (= empty word)

while $i \leq n$ **do**
    if $a_i \neq a$ **then**
        $w := wa_i$;
        $i := i + 1$
    else
        $j :=$ undefined;
        for $k = i + 1$ **to** n **do**
            if $a_k = a$ and $\widehat{\alpha}(a_{i+1} \cdots a_{k-1}) \subseteq I(a)$ **then**
                $j := k$
            endif
        endfor
        if $j =$ undefined **then**
            $w := wa_i$;
            $i := i + 1$
        else
            $w := w \, \pi_{\Sigma \setminus \{a\}}(a_i \cdots a_{j-1})$;
            $i := j + 1$
        endif
    endif
endwhile
return(w)   (= the content of the output tape of the transducer)

Let w_s be the content of the output tape at the beginning of stage s, i.e., when the algorithm checks the condition of the while-loop and variable i has value s. (hence, $w_1 = 1$ and w_{n+1} is the final output). The invariant of the algorithm is that

- $w_s = a_1 \cdots a_{s-1}$ in $C(\Sigma, I)$,
- w_s is a-short, and
- if $a_1 \cdots a_{s-1}$ is b-short, then also w_s is b-short.

The proof of this fact uses Lemma 5.1. $\square$

6. Free partially commutative inverse monoids

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

$$(6.1) \qquad x\bar{x}x = x, \quad \bar{x}x\bar{x} = \bar{x}, \text{ and } \quad x\bar{x}\,y\bar{y} = y\bar{y}\,x\bar{x}.$$

The element $\bar{x}$ is uniquely defined by these properties and it is called the *inverse* of x. Thus, we may also use the notation $\bar{x} = x^{-1}$. 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 (6.1) 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 [**28**].

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

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

It is an inverse monoid over (Σ, I). Da Costa has studied $\mathrm{FIM}(\Sigma, I)$ in his PhD thesis [**39**]. 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 [15], 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 $FIM(\Sigma, I)$ is very low, too.

THEOREM 6.1. *The word problem of* $FIM(\Sigma, I)$ *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 the following subset of the trace monoid $M(\Gamma, I_\Gamma)$.

$$(6.2) \qquad M(u) = \bigcup_{i=1}^{n} \{p \mid p \text{ is a prime prefix of } \widehat{u_i}\} \subseteq M(\Gamma, I_\Gamma).$$

(This set is a partial commutative analogue of the classical notion of *Munn tree* introduced in [34].) It is shown in [15, Sect. 3] that for all words $u, v \in \Gamma^*$, $u = v$ in $FIM(\Sigma, I)$ if and only if

(a) $u = v$ in the graph group $G(\Sigma, I)$ and
(b) $M(u) = M(v)$.

Since $G(\Sigma, I)$ is linear, condition (a) can be checked in logspace [29, 38]. For (b), it suffices to show that the set $M(u)$ from (6.2) 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 Theorem 5.4 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 Lemma 4.1), 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 able to compute in logspace the length of the shortlex normal form and the set of letters appearing in the shortlex normal form. For even Coxeter groups we can do better and enhance the general result since we can compute the Parikh-image of geodesics. An obvious open problem is, whether for every (even) 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 [31], 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 [31]. So, one might first try to lower this bound e.g. to $\mathsf{LOGDCFL}$ (the class of all languages that are logspace-reducible to a deterministic context-free language).

References

1. Carme Àlvarez and Joaquim Gabarró, *The parallel complexity of two problems on concurrency*, Inform. Process. Lett. **38** (1991), no. 2, 61–70.
2. Mladen Bestvina and Noel Brady, *Morse theory and finiteness properties of groups*, Inventiones Mathematicae **129** (1997), no. 3, 445–470.

3. Anders Björner and Francesco Brenti, *Combinatorics of Coxeter groups*, Graduate Texts in Mathematics, vol. 231, Springer, New York, 2005. MR MR2133266 (2006d:05001)

4. Ron Book and Friedrich Otto, *String-rewriting systems*, Springer-Verlag, 1993.

5. Brigitte Brink and Robert B. Howlett, *A finiteness property and an automatic structure for Coxeter groups*, Math. Ann. **296** (1993), 179–190.

6. Jin-Yi Cai, *Parallel computation over hyperbolic groups*, Proc. 24th ACM Symp. on Theory of Computing, STOC 92, ACM-press, 1992, pp. 106–115.

7. William A. Casselman, *Automata to perform basic calculations in Coxeter groups*, Representations of groups (Banff, AB, 1994), CMS Conf. Proc., vol. 16, Amer. Math. Soc., Providence, RI, 1995, pp. 35–58.

8. Andrew Chiu, George Davida, and Bruce Litow, *Division in logspace-uniform* NC^1, Theoretical Informatics and Applications. Informatique Théorique et Applications **35** (2001), no. 3, 259–275.

9. John Crisp, Eddy Godelle, and Bert Wiest, *The conjugacy problem in right-angled Artin groups and their subgroups*, Journal of Topology **2** (2009), no. 3, 442460.

10. John Crisp and Bert Wiest, *Embeddings of graph braid and surface groups in right-angled Artin groups and braid groups*, Algebraic & Geometric Topology **4** (2004), 439–472.

11. Samir Datta and Rameshwar Pratap, *Computing bits of algebraic numbers*, Tech. report, arXiv.org, 2011, `http://arxiv.org/abs/1112.4295`.

12. Michael W. Davis, *The geometry and topology of Coxeter groups*, London Math. Soc. Monographs Series, vol. 32, Princeton University Press, Princeton, NJ, 2008. MR MR2360474 (2008k:20091)

13. Volker Diekert, *Combinatorics on traces*, Lecture Notes in Computer Science, no. 454, Springer-Verlag, Heidelberg, 1990.

14. Volker Diekert, Jonathan Kausch, and Markus Lohrey, *Logspace computations in graph groups and Coxeter groups*, Lecture Notes in Computer Science, Springer-Verlag, 2012, *To appear in Proc. LATIN'2012, Arequipa, Peru.*

15. Volker Diekert, Markus Lohrey, and Alexander Miller, *Partially commutative inverse monoids*, Semigroup Forum **77** (2008), no. 2, 196–226.

16. Volker Diekert and Anca Muscholl, *Solvability of equations in free partially commutative groups is decidable*, International Journal of Algebra and Computation **16** (2006), 1047–1070, Journal version of ICALP 2001, 543–554, LNCS 2076.

17. Volker Diekert and Grzegorz Rozenberg (eds.), *The book of traces*, World Scientific, Singapore, 1995.

18. Carl Droms, Jacques Lewin, and Herman Servatius, *The length of elements in free solvable groups*, Proc. Amer. Math. Soc. **119** (1993), 27–33.

19. Murray Elder, *A linear-time algorithm to compute geodesics in solvable Baumslag-Solitar groups*, Illinois Journal of Mathematics **54** (2010), no. 1, 109–128.

20. Murray Elder, Gillian Elston, and Gretchen Ostheimer, *On groups that have normal forms computable in logspace*, Tech. report, arXiv.org, 2012, `http://arxiv.org/abs/1201.4363`.

21. Murray Elder and Andrew Rechnitzer, *Some geodesic problems in groups*, Groups. Complexity. Cryptology **2** (2010), no. 2, 223–229.

22. David B. A. Epstein, James W. Cannon, Derek F. Holt, Silvio V. F. Levy, Michael S. Paterson, and William P. Thurston, *Word processing in groups*, Jones and Bartlett, Boston, 1992.

23. R. Ghrist and V. Peterson, *The geometry and topology of reconfiguration*, Advances in Applied Mathematics **38** (2007), no. 3, 302–323.

24. William Hesse, Eric Allender, and David A. Mix Barrington, *Uniform constant-depth threshold circuits for division and iterated multiplication*, Journal of Computer and System Sciences **65** (2002), 695–716.

25. Tim Hsu and Daniel T. Wise, *On linear and residual properties of graph products*, Michigan Mathematical Journal **46** (1999), no. 2, 251–259.

26. Emil Jeřábek, *Root finding with threshold circuits*, Tech. report, arXiv.org, 2011, `http://arxiv.org/abs/1112.4295`.

27. Olga Kharlampovich and Atefeh Mohajeri Moghaddam, *Approximation of geodesics in metabelian groups*, International Journal of Algebra and Computation **22** (2012), no. 2.

28. Mark V. Lawson, *Inverse semigroups: The theory of partial symmetries*, World Scientific, 1999.

29. Richard J. Lipton and Yechezkel Zalcstein, *Word problems solvable in logspace*, Journal of the Association for Computing Machinery **24** (1977), no. 3, 522–526.
30. Bruce E. Litow, *On sums of roots of unity*, Proceedings of ICALP 2010, Lecture Notes in Computer Science, vol. 6198, Springer, 2010, pp. 420–425.
31. Markus Lohrey, *Decidability and complexity in automatic monoids*, International Journal of Foundations of Computer Science **16** (2005), no. 4, 707–722.
32. Markus Lohrey and Nicole Ondrusch, *Inverse monoids: Decidability and complexity of algebraic questions*, Inf. Comput. **205** (2007), 1212–1234.
33. Charles F. Miller III, *Decision problems for groups – survey and reflections*, Algorithms and Classification in Combinatorial Group Theory, Springer, 1992, pp. 1–60.
34. Walter D. Munn, *Free inverse semigroups*, Proc. London Math. Soc. **29** (1974), no. 3, 385–404.
35. Alexei Myasnikov, Vitalii Roman'kov, Alexander Ushakov, and Anatoly Vershik, *The word and geodesic problems in free solvable groups*, Transactions of the American Mathematical Society **362** (2010), 4655–4682.
36. Christos Papadimitriou, *Computation complexity*, Addison-Wesley, 1994.
37. Michael Paterson and Alexander Razborov, *The set of minimal braids is co-NP-complete*, J. Algorithms **12** (1991), 393–408.
38. Hans-Ulrich Simon, *Word problems for groups and contextfree recognition*, Proceedings of Fundamentals of Computation Theory (FCT'79), Berlin/Wendisch-Rietz (GDR), Akademie-Verlag, 1979, pp. 417–422.
39. António Augusto Veloso da Costa, *γ-produtos de monóides e semigrupos*, Ph.D. thesis, Universidade do Porto, Faculdade de Ciências, 2003.
40. Heribert Vollmer, *Introduction to circuit complexity*, Springer, Berlin, 1999.
41. Stephan Waack, *Tape complexity of word problems*, Proceedings of Fundamentals of Computation Theory (FCT'81) (Ferenc Gécseg, ed.), Lecture Notes in Computer Science, vol. 117, Springer, 1981, pp. 467–471.

FMI, Universität Stuttgart, Germany

FMI, Universität Stuttgart, Germany

Insitut für Informatik, Universität Leipzig, Germany