



HAL
open science

Algorithms for Garside calculus

Patrick Dehornoy, Volker Gebhardt

► **To cite this version:**

| Patrick Dehornoy, Volker Gebhardt. Algorithms for Garside calculus. 2013. hal-00776088

HAL Id: hal-00776088

<https://hal.science/hal-00776088>

Preprint submitted on 15 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ALGORITHMS FOR GARSIDE CALCULUS

PATRICK DEHORNOY AND VOLKER GEBHARDT

ABSTRACT. Garside calculus is the common mechanism that underlies a certain type of normal form for the elements of a monoid, a group, or a category. Originating from Garside's approach to Artin's braid groups, it has been extended to more and more general contexts, the latest one being that of categories and what are called Garside families. One of the benefits of this theory is to lead to algorithms solving effectively the naturally occurring problems, typically the Word Problem. The aim of this paper is to present and solve these algorithmic questions in the new extended framework.

In 1969, F.A. Garside [20] solved the Word and Conjugacy Problems in Artin's braid group B_n [2] by describing the latter as a group of fractions and analyzing the involved monoid in terms of its divisibility relation. This approach was continued and extended in several steps, first to Artin-Tits groups of spherical type [3, 16, 1, 24, 19, 18, 4], then to a larger family of groups now known as Garside groups [10, 8]. More recently, it was realized that going to a categorical context allows for capturing further examples [17], and a coherent theory has recently emerged with a central unifying notion called Garside families [14, 15]: the central notion is a certain way of decomposing the elements of the reference category or its groupoid of fractions and a Garside family is what makes the construction possible.

What we do in this paper is to present and analyse the main algorithms arising in this new, extended context of Garside families, with two main directions, namely recognizing that a candidate family is a Garside family and using a Garside family to compute in the category, typically finding distinguished decompositions and solving the Word Problem along the lines of [11]. This results in a corpus of about twenty algorithms that are proved to be correct, analysed, and given examples. We do not address the Conjugacy Problem here, as extending the methods of [21, 22] will require further developments that we keep for a subsequent work.

The paper consists of five sections. Section 1 is a review of Garside families and the derived notions involved in the approach, together with some basic results that appear in other sources. Next, we address the question of effectively recognizing Garside families and we describe and analyse algorithms doing it: in Section 2, we consider the case when the ambient category is specified using a presentation (of a certain type), whereas, in Section 3, we consider the alternative approach when the category is specified using what is called a germ. Finally, the last two sections are devoted to those computations that can be developed once a Garside family is given. In Section 4, we consider computations taking place in the reference category or monoid ("positive case"), whereas, in Section 5, we address similar questions in the groupoid or group of fractions of the reference category ("signed case").

1. THE GENERAL CONTEXT

In this introductory section, we present the background of categories and Garside families, together with some general existence and uniqueness results that will be used and, often, refined in the sequel of the paper. Proofs appear in other sources and they will be omitted in general.

1991 *Mathematics Subject Classification.* 20F10, 18B40, 20F36, 68Q17.

Key words and phrases. normal form, word problem, category presentation, groupoid of fractions, Garside calculus, Garside family, germ.

Both authors acknowledge support under Australian Research Council's Discovery Projects funding scheme (project number DP1094072). Volker Gebhardt acknowledges support under the Spanish Project MTM2010-19355.

1.1. Categories. The general context is that of categories, which should be seen here just as monoids with a partial product, that is, one that is not necessarily defined everywhere. A *pre-category* (or multigraph) is a family \mathcal{A} plus two maps, “source” and “target”, of \mathcal{A} to another family $\text{Obj}(\mathcal{A})$ (the objects of \mathcal{A}), and a *category* is a precategory equipped with a partial multiplication such that fg exists if and only if the target of f , denoted $\text{trg}(f)$, coincides with the source of g , denoted $\text{src}(g)$. The multiplication is associative whenever defined and, in addition, has a neutral element 1_x for each object x , that is, $1_x g = g = g 1_y$ holds for every g with source x and target y . If \mathcal{C} is a category, the family of all neutral elements is denoted by $\mathbf{1}_{\mathcal{C}}$ and, for \mathcal{A} included in \mathcal{C} and x, y in $\text{Obj}(\mathcal{C})$, the family of elements of \mathcal{A} with source x and target y is denoted by $\mathcal{A}(x, y)$. A monoid is the special case of a category when there is only one object, so that the product is always defined. It is convenient to represent the elements of a category using arrows, so that an element g with source x and target y is represented by $x \xrightarrow{g} y$.

Free categories and paths. If \mathcal{A} is a precategory, the free category generated by \mathcal{A} is the family \mathcal{A}^* of all \mathcal{A} -paths, that is, all finite sequences (a_1, \dots, a_p) of elements of \mathcal{A} such that the target of a_{i-1} is the source of a_i for every i , together with, for each object x , an empty path ε_x , and equipped with concatenation of paths. When w is an \mathcal{A} -path, we denote by $\|w\|$ the length of w , and, for $1 \leq i \leq \|w\|$, we denote by $w[i]$ the i -th entry in w . For \mathcal{B} included in \mathcal{A} , we denote the family of all \mathcal{B} -paths of length p by $\mathcal{B}^{[p]}$. If w_1, w_2 are two paths, we denote by $w_1|w_2$ the concatenation of w_1 and w_2 when it exists, that is, when the target of w_1 (defined to be the target of the last entry in w_1) coincides with the source of w_2 (defined to be the source of the first entry in w_2). We identify a length one path with its unique entry. Then a length p path (a_1, \dots, a_p) is the concatenation of the length one paths made of its successive entries, so that it can be denoted by $a_1|\dots|a_p$. When \mathcal{A} is a set, that is, a precategory with one object only, the condition about source and targets vanishes, and it is usual to say \mathcal{A} -word for \mathcal{A} -path.

Presentations. Every category that is generated by a family \mathcal{A} is a quotient of the free category \mathcal{A}^* , and, for \mathcal{R} a family of pairs of \mathcal{A} -paths, it is said to admit the presentation $(\mathcal{A}; \mathcal{R})$ if it is isomorphic to $\mathcal{A}^*/\equiv^+_{\mathcal{R}}$ where $\equiv^+_{\mathcal{R}}$ is the congruence on \mathcal{A}^* generated by \mathcal{R} . The elements of \mathcal{R} are called *relations*, and, in this context, it is customary to write $u = v$ instead of (u, v) for a relation and, in concrete examples, to omit the concatenation sign, thus writing $a_1 \dots a_p = b_1 \dots b_q$ rather than $a_1|\dots|a_p = b_1|\dots|b_q$. If $(\mathcal{A}; \mathcal{R})$ is a presentation, we write $\langle \mathcal{A} | \mathcal{R} \rangle^+$ for the category presented by $(\mathcal{A}; \mathcal{R})$ —which is determined only up to isomorphism—and, for w an \mathcal{A} -path, we write $[w]$ for the $\equiv^+_{\mathcal{R}}$ -class of w , that is, for the element of $\langle \mathcal{A} | \mathcal{R} \rangle^+$ represented by w —which is also the evaluation of the path w in the category $\langle \mathcal{A} | \mathcal{R} \rangle^+$.

Cancellativity. All categories we shall consider here will have to satisfy some cancellability condition, at least on one side.

Definition 1.1. A category \mathcal{C} is called *left-cancellative* (resp. *right-cancellative*) if $fg = fg'$ (resp. $gf = g'f$) implies $g = g'$ for all f, g, g' in \mathcal{C} .

A category is called *cancellative* if it is both left- and right-cancellative. In a left- (or right-) cancellative category, an element has a left-inverse if and only if it has a right-inverse, and so there is a unique, non-ambiguous notion of *invertible* element. For \mathcal{C} a left-cancellative category, we denote by \mathcal{C}^\times the subgroupoid of \mathcal{C} consisting of all invertible elements. An invertible element will be called *nontrivial* if it is not an identity-element 1_x .

Divisibility, lcm, lcm-selector. Associated with every (left-cancellative) category—hence, in particular, every monoid—comes a natural left-divisibility relation.

Definition 1.2. Assume that \mathcal{C} is a left-cancellative category. For f, g in \mathcal{C} , we say that f is a *left-divisor* of g , or, equivalently, that g is a *right-multiple* of f , denoted $f \preceq g$, if there exists g' in \mathcal{C} so that $fg' = g$ holds.

The hypothesis that the ambient category is left-cancellative is needed to guarantee that left-divisibility is a partial preordering; the associated equivalence relation is right-multiplication by an invertible element: the conjunction of $f \preceq g$ and $g \preceq f$ is equivalent to the existence of an invertible element e satisfying $fe = g$, which will be denoted by $f =^{\times} g$ hereafter.

Simple derived notions stem from the left-divisibility relation, corresponding to greatest common lower bound and least common upper bound. We say that h is a *greatest common left-divisor*, or *left-gcd* of f and g if h left-divides f and g and every left-divisor of f and g left-divides h . Symmetrically, we say h is a *least common right-multiple*, or *right-lcm* of f and g if h is a right-multiple of f and g and every right-multiple of f and g is a right-multiple of h . Right-lcms and left-gcds, if they exist, are unique up to right-multiplication by an invertible element, hence unique if the ambient category has no nontrivial invertible element. In the latter case, we write $f \vee g$ and $f \wedge g$ for the right-lcm and the left-gcd of f and g (if they exist), and $f \setminus g$ for the unique element that satisfies $f \vee g = f(f \setminus g)$.

Definition 1.3. We say that a left-cancellative category \mathcal{C} *admits right-lcms* (resp. *admits local right-lcms*) if any two elements of \mathcal{C} with the same source (resp. if any two elements of \mathcal{C} that admit a common right-multiple) admit a right-lcm.

Finally, we shall sometimes need to choose right-lcms explicitly. The following terminology is then natural.

Definition 1.4. Assume that \mathcal{C} is a left-cancellative category, and \mathcal{A} is a generating subfamily of \mathcal{C} . A *right-lcm selector* on \mathcal{A} is a partial map $\theta : \mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}^*$ such that, for all a, b in \mathcal{A} , the elements $\theta(a, b)$ and $\theta(b, a)$ are defined if and only if a and b admit a right-lcm and, in this case, there exists a right-lcm h of a and b such that both $a|\theta(a, b)$ and $b|\theta(b, a)$ represent h .

Note that, if \mathcal{C} is a left-cancellative category that admits no nontrivial invertible element, then the map $(f, g) \mapsto f \setminus g$ is a right-lcm selector on \mathcal{C} .

The notions of a right-divisor and left-multiple are defined symmetrically, and so are the derived notions of a right-gcd, and a left-lcm selector.

Noetherianity conditions.

Definition 1.5. A left-cancellative category \mathcal{C} is called *right-Noetherian* if every bounded \prec -increasing sequence in \mathcal{C} is finite.

By extension, we say that a (positive) presentation is right-Noetherian if the associated category is right-Noetherian. Standard results (see for instance [23]) give the following criterion for establishing Noetherianity conditions.

Lemma 1.6. *A presentation $(\mathcal{A}; \mathcal{R})$ is right-Noetherian if and only if there exists a map λ of \mathcal{A}^* to the ordinals that is $\equiv^+_{\mathcal{R}}$ -invariant and satisfies $\lambda(a) > 0$ for every a in \mathcal{A} and $\lambda(a|w) > \lambda(w)$ for all w in \mathcal{A}^* and a in \mathcal{A} .*

Note that every presentation $(\mathcal{A}; \mathcal{R})$ such that \mathcal{R} consists of relations of the form $u = v$ with u, v of the same length is right-Noetherian, as one can then define $\lambda(w)$ to be the length of w .

Ore category, groupoid of fractions. Finally, we recall that a groupoid is a category in which every element is invertible, a group corresponding to the special case when there is only one object, that is, the product is always defined.

We say that a groupoid \mathcal{G} is a *groupoid of left-fractions* for a subcategory \mathcal{C} if every element of \mathcal{G} admits an expression of the form $f^{-1}g$ with f, g in \mathcal{C} . The following result of Ore is classical:

Proposition 1.7. [5] *Say that a category is left-Ore (resp. right-Ore) if it is cancellative and any two elements that have the same target (resp. source) admit a common left-multiple (resp. right-multiple). Then a category embeds in a groupoid of left-fractions (resp. of right fractions) if and only if it is left-Ore (resp. right-Ore).*

A category that is both left- and right-Ore will be called an *Ore category*.

1.2. Normal decompositions and Garside families. The central idea in our approach consists in introducing distinguished decompositions of a certain type for the elements of the considered category. These decompositions involve a reference subfamily of the ambient category and correspond to the principle of recursively selecting maximal left-divisors lying in the reference family.

Definition 1.8. Assume that \mathcal{C} is a left-cancellative category. For \mathcal{S} included in \mathcal{C} , a \mathcal{C} -path $g_1|\cdots|g_p$ is called \mathcal{S} -greedy (resp. \mathcal{S} -normal) if, for every $i < p$, we have

$$(1.9) \quad \forall a \in \mathcal{S} \quad \forall f \in \mathcal{C} \quad (a \preceq f g_i g_{i+1} \Rightarrow a \preceq f g_i)$$

(resp. this and, in addition, every entry a_i lies in \mathcal{S}^\sharp , defined to be $\mathcal{S}^\times \cup \mathcal{C}^\times$).

When using diagrams in which the elements of the category are represented by arrows, we shall indicate that a path $g_1|g_2$ is \mathcal{S} -greedy by appending a small arc as in $\xrightarrow{g_1} \overset{\curvearrowright}{} \xrightarrow{g_2} \cdot$.

Example 1.10. Consider the free Abelian monoid M defined by the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{ab} = \mathbf{ba})$ and let $\mathcal{S} = \{\mathbf{a}, \mathbf{b}, \mathbf{ab}\}$, whence $\mathcal{S}^\sharp = \{1, \mathbf{a}, \mathbf{b}, \mathbf{ab}\}$. The \mathcal{S} -normal paths in M are precisely the paths $g_1|\cdots|g_q$ that, for some p, p' in $\{0, \dots, q\}$, satisfy (i) $g_i = \mathbf{ab}$ for $i = 1, \dots, p$; (ii) either $g_i = \mathbf{a}$ for $i = p+1, \dots, p'$, or $g_i = \mathbf{b}$ for $i = p+1, \dots, p'$; and (iii) $g_i = 1$ for $i = p'+1, \dots, q$.

If g is an element of a category \mathcal{C} , a path w satisfying $[w] = g$ is called a *decomposition* of g . What we shall be interested in in the sequel are the (possible) \mathcal{S} -normal decompositions of the elements of the considered category. One of the interests of such decompositions is that they are essentially unique.

Proposition 1.11. [14, Proposition 2.11] *Assume that \mathcal{C} is a left-cancellative category and \mathcal{S} is included in \mathcal{C} . Then any two \mathcal{S} -normal decompositions of an element of \mathcal{C} (if any) are \mathcal{C}^\times -deformations of one another, where $a_1|\cdots|a_p$ is said to be a \mathcal{C}^\times -deformation of $b_1|\cdots|b_q$ if there exist invertible elements e_0, \dots, e_r , with $r = \max(p, q)$, such that e_0, e_r are identity-elements and $b_i e_i = e_{i-1} a_i$ holds for $1 \leq i \leq r$, where, for $p \neq q$, the shorter path is expanded by identity-elements (see Figure 1).*

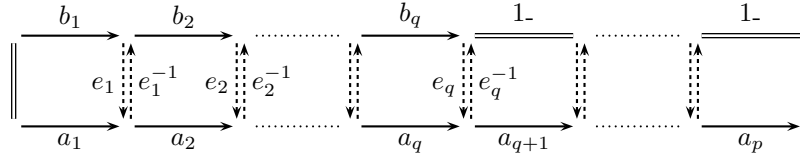


FIGURE 1. Deformation by invertible elements: invertible elements connect the corresponding entries; if one path is shorter than the other (here we are in the case $q < p$), it is extended by identity-elements.

Note that, if \mathcal{C} contains no nontrivial invertible element, that is, the only invertible elements are the identity-elements, then Proposition 1.11 provides a genuine uniqueness result provided one discards the \mathcal{S} -normal paths that finish with an identity-element.

As for the existence of \mathcal{S} -normal decompositions, it naturally depends on the family \mathcal{S} . Here is where Garside families appear:

Definition 1.12. A subfamily \mathcal{S} of a left-cancellative category \mathcal{C} is called a *Garside family* if every element of \mathcal{C} admits an \mathcal{S} -normal decomposition.

Every left-cancellative category is a Garside family in itself, so every left-cancellative category contains a Garside family. Practically recognizing whether a given family is a Garside family will be one of the main tasks of Sections 2 and 3 below. We shall appeal to the following simple characterization which is valid whenever the ambient category satisfies special assumptions:

Lemma 1.13. [15, Corollary IV.2.18] *Assume that \mathcal{C} is a left-cancellative category that is right-Noetherian and admits unique local right-lcms. Then a subfamily \mathcal{S} of \mathcal{C} is a Garside family in \mathcal{C} if and only if \mathcal{S} generates \mathcal{C} and, for all a, b in \mathcal{S} admitting a common right-multiple, $a \vee b$ and $a \setminus b$ lie in $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$.*

Also, we shall use the following closure result, which in some sense extends Lemma 1.13, but need not characterize Garside families in general.

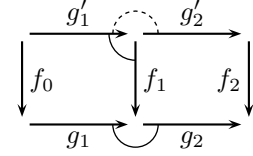
Lemma 1.14. [14, Proposition 3.9] *If \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} and c is a common right-multiple of two elements a, b of \mathcal{S}^\sharp , there exists a common right-multiple c' of a and b such that c is a right-multiple of c' and c' , together with a' and b' defined by $ab' = ba' = c'$, lie in \mathcal{S}^\sharp .*

An application of Lemma 1.14 is that every Garside family gives rise to a simple presentation of the ambient category. For our current purpose it will be sufficient to state the result in the particular case when no nontrivial invertible element exists, so that a right-lcm is unique when it exists.

Proposition 1.15. [15, Proposition IV.3.6] *Assume that \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} that contains no nontrivial invertible element and admits local right-lcms. Let \mathcal{R} consist of all relations $a(a \setminus b) = b(b \setminus a)$ for a, b in \mathcal{S} admitting a common right-multiple. Then $(\mathcal{S}; \mathcal{R})$ is a presentation of \mathcal{C} .*

Finally, we shall use the following simple diagrammatic rule about \mathcal{S} -greedy paths.

Lemma 1.16. [14, Lemma 3.3] *(first domino rule) Assume that \mathcal{C} is a left-cancellative category, \mathcal{S} is included in \mathcal{C} , and we have a commutative diagram with edges in \mathcal{C} as on the right. If $g_1|g_2$ and $g'_1|f_1$ are \mathcal{S} -greedy, then $g'_1|g'_2$ is \mathcal{S} -greedy as well.*



A second, symmetric domino rule will be mentioned in Lemma 1.31 below.

1.3. Symmetric normal decompositions and strong Garside families. If \mathcal{C} is a left-Ore category, then, by Proposition 1.7, it embeds in the groupoid of left-fractions that we shall denote by $\text{Env}(\mathcal{C})$ (like “enveloping groupoid”). If \mathcal{S} is a Garside family in \mathcal{C} , every element of \mathcal{C} admits an \mathcal{S} -normal decomposition, and it is natural to try to extend the result from \mathcal{C} to $\text{Env}(\mathcal{C})$, that is, to find distinguished decompositions for the elements of $\text{Env}(\mathcal{C})$ in terms of elements of \mathcal{S}^\sharp and their inverses.

To do it, we extend the notion of an \mathcal{A} -path into that of a signed \mathcal{A} -path. Formally, if \mathcal{A} is any precategory, we introduce a family $\overline{\mathcal{A}}$ that is disjoint from and in bijection to \mathcal{A} as $\overline{\mathcal{A}} = \{\overline{a} \mid a \in \mathcal{A}\}$, where the source of \overline{a} is the target of a and vice versa. A *signed \mathcal{A} -path* is defined to be a $(\mathcal{A} \cup \overline{\mathcal{A}})$ -path. We extend the “bar” map to all signed paths by defining $\overline{\overline{a}} = a$ for a in \mathcal{A} and $\overline{w_1|w_2} = \overline{w_2}|\overline{w_1}$. For \mathcal{G} a groupoid and \mathcal{A} included in \mathcal{G} , we extend the notation $[w]$ to signed \mathcal{A} -paths by declaring that $[\overline{g}]$ is g^{-1} for g in \mathcal{A} , that is, we use the letters of $\overline{\mathcal{A}}$ to represent the inverses of the elements of \mathcal{A} . If w is a signed path and $[w] = g$ holds, we again say that w is a decomposition of g .

When \mathcal{C} is a left-Ore category and \mathcal{S} is a Garside family of \mathcal{C} , we now look for distinguished decompositions for the elements of the groupoid $\text{Env}(\mathcal{C})$. As every element of $\text{Env}(\mathcal{C})$ is a left-fraction, it is natural to consider decompositions that are *negative–positive \mathcal{S}^\sharp -paths*, this meaning that every negative entry precedes every positive entry, where the entries in \mathcal{S}^\sharp are called positive and those in $\overline{\mathcal{S}^\sharp}$ are called negative.

Definition 1.17. Assume that \mathcal{C} is a left-Ore category.

(i) Two elements f, g of \mathcal{C} are called *left-disjoint* if, for all f', g' in \mathcal{C} satisfying $f^{-1}g = f'^{-1}g'$ in $\text{Env}(\mathcal{C})$, there exists h in \mathcal{C} satisfying $f' = hf$ and $g' = hg$.

(ii) For \mathcal{S} included in \mathcal{C} , a negative–positive path $\overline{g_q}|\dots|\overline{g_1}|f_1|\dots|f_p$ is called *symmetric \mathcal{S} -greedy* (resp. *symmetric \mathcal{S} -normal*) if $g_1|\dots|g_q$ and $f_1|\dots|f_p$ are \mathcal{S} -greedy (resp. \mathcal{S} -normal) and, in addition, g_1 and f_1 are left-disjoint.

When using diagrams, we shall indicate that two elements f, g are left-disjoint by appending a small arc as in $\overleftarrow{g} \overbrace{\quad}^f \rightarrow$. So a generic symmetric \mathcal{S} -greedy path pictorially corresponds to a diagram



and it is symmetric \mathcal{S} -normal if, in addition, all edges correspond to elements of \mathcal{S}^\sharp . Note that a positive path is symmetric \mathcal{S} -normal if and only if it is \mathcal{S} -normal: indeed, a positive path is a negative–positive path whose negative part is empty, and, for every g in $\mathcal{S}^\sharp(x, -)$, the elements ε_x and g are (trivially) left-disjoint.

Example 1.18. Consider the free Abelian monoid \mathcal{A} and the set \mathcal{S} as in Example 1.10. The unordered pairs of left-disjoint elements of \mathcal{S}^\sharp are precisely $\{1, 1\}$, $\{1, \mathbf{a}\}$, $\{1, \mathbf{b}\}$, $\{1, \mathbf{ab}\}$, and $\{\mathbf{a}, \mathbf{b}\}$, where $1 = 1_x$ for the unique element x of $\text{Obj}(\mathcal{A})$.

Like \mathcal{S} -normal decompositions in the positive case, symmetric \mathcal{S} -normal decompositions turn out to be (nearly) unique when they exist.

Proposition 1.19. [15, Proposition III.2.16] *Assume that \mathcal{C} is a left-Ore category and \mathcal{S} is included in \mathcal{C} . Then any two symmetric \mathcal{S} -normal decompositions of an element of $\text{Env}(\mathcal{C})$ (if any) are \mathcal{C}^\times -deformations of one another, this meaning that there exists a commutative diagram as in Figure 2.*

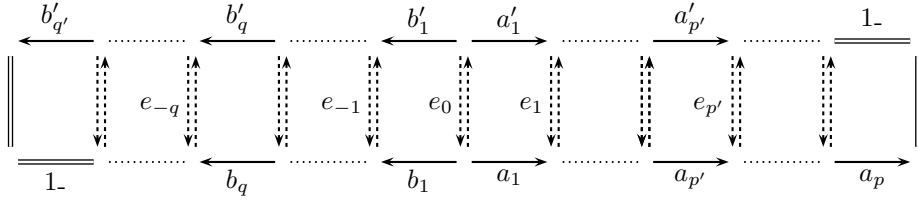


FIGURE 2. Deformation of a signed path by invertible elements: invertible elements connect the corresponding entries; if some path is shorter than the other (here we have $q < q'$ and $p' < p$), it is extended by identity-elements.

As for existence, symmetric normal decompositions are connected with left-lcms:

Lemma 1.20. [15, Lemma III.2.19] *Assume that \mathcal{S} is a Garside family in a left-Ore category \mathcal{C} , that $a_1 | \dots | a_p$ and $b_1 | \dots | b_q$ are \mathcal{S} -normal paths, and that $b_1 | \dots | b_q | g = a_1 | \dots | a_p | f$ holds. Then $\overline{b_q} | \dots | \overline{b_1} | a_1 | \dots | a_p$ is symmetric \mathcal{S} -greedy if and only if $b_1 \dots b_q$ and $a_1 \dots a_p$ are left-disjoint, if and only if $b_1 \dots b_q g$ is a left-lcm of f and g .*

Lemma 1.20 implies that an element fg^{-1} of $\text{Env}(\mathcal{C})$ admits a symmetric \mathcal{S} -normal decomposition if and only if f and g admit a left-lcm in \mathcal{C} , whence:

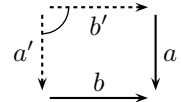
Proposition 1.21. *If \mathcal{S} is a Garside family in a left-Ore category \mathcal{C} that admits left-lcms, every element of $\text{Env}(\mathcal{C})$ that can be expressed as a right-fraction admits a symmetric \mathcal{S} -normal decomposition.*

In particular, if \mathcal{C} is an Ore category, then every element of $\text{Env}(\mathcal{C})$ can be expressed as a right-fraction, and we obtain:

Corollary 1.22. *If \mathcal{S} is a Garside family in an Ore category \mathcal{C} that admits left-lcms, every element of $\text{Env}(\mathcal{C})$ admits a symmetric \mathcal{S} -normal decomposition.*

When algorithmic questions are involved, it will be convenient to consider special Garside families that we introduce now.

Definition 1.23. Assume that \mathcal{C} is a left-Ore category. A Garside family \mathcal{S} of \mathcal{C} is called *strong* if, for all a, b in \mathcal{S}^\sharp with the same target, there exist a', b' in \mathcal{S}^\sharp that are left-disjoint and satisfy $a'b = b'a$.



The interest of introducing the notion of a strong Garside family is to allow for a refined version of Proposition 1.21:

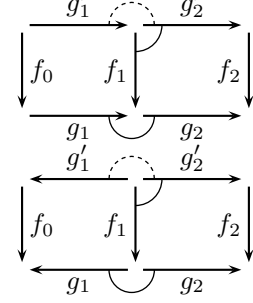
Proposition 1.24. [15, Proposition III.2.31] *If \mathcal{S} is a strong Garside family in a left-Ore category \mathcal{C} that admits left-lcms, every element of $\text{Env}(\mathcal{C})$ that admits a positive-negative \mathcal{S}^\sharp -decomposition of length ℓ admits a symmetric \mathcal{S} -normal decomposition of length at most ℓ .*

Finally, as in the positive case, we shall appeal to diagrammatic rules involving normal paths.

Lemma 1.25. [15, Propositions III.2.39 and III.2.42]

(i) *(third domino rule) Assume that \mathcal{C} is a left-cancellative category, \mathcal{S} is included in \mathcal{C} , and we have a commutative diagram with edges in \mathcal{C} as on the right. If $g_1|g_2$ is \mathcal{S} -greedy, and f_1, g'_2 are left-disjoint, then $g'_1|g'_2$ is \mathcal{S} -greedy as well.*

(ii) *(fourth domino rule) Assume that \mathcal{C} is a left-Ore category, \mathcal{S} is included in \mathcal{C} , and we have a commutative diagram with edges in \mathcal{C} as on the right. If g_1, g_2 are left-disjoint, and f_1, g'_2 are left-disjoint, then g'_1, g'_2 are left-disjoint as well.*



1.4. Bounded Garside families and Δ -normal decompositions. In many cases, interesting Garside families in a monoid consist of the left-divisors of some maximal element Δ , in which case it is natural to call them bounded by Δ . In a category context, the maximal element has to depend on the source, and it is replaced with a map from the objects to the elements.

Definition 1.26. A Garside family \mathcal{S} in a cancellative category \mathcal{C} is called *bounded* if there exists a map Δ from $\text{Obj}(\mathcal{C})$ to \mathcal{C} satisfying the following conditions:

- (i) $a \in \mathcal{S}(x, -)$ implies $a \preceq \Delta(x)$,
- (ii) $\forall y \in \text{Obj}(\mathcal{C}) \exists! z \in \text{Obj}(\mathcal{C}) \forall a \in \mathcal{S}^\sharp(-, y) \exists a' \in \mathcal{S}^\sharp(z, -) (a'a = \Delta(z))$.

Note that, in Definition 1.26, (i) and (ii) are symmetric, as (i) can be stated as

$$\forall x \in \text{Obj}(\mathcal{C}) \exists! y \in \text{Obj}(\mathcal{C}) \forall a \in \mathcal{S}^\sharp(x, -) \exists a' \in \mathcal{S}^\sharp(-, y) (aa' = \Delta(x)),$$

with y the target of $\Delta(x)$. In the above context, for every element a of \mathcal{C} , the unique element a' satisfying $aa' = \Delta(x)$ is denoted by $\partial(a)$, whereas the unique element a' satisfying $a'a = \Delta(z)$ for some z is denoted by $\tilde{\partial}(a)$.

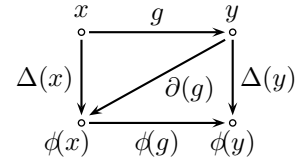
Remark 1.27. A Garside family satisfying Definition 1.26(i) only is called *right-bounded* by Δ . The latter notion is natural and useful in the positive case, but it is not sufficient in the signed case and we shall not consider it here.

Proposition 1.28. [15, Proposition VI.3.11] (i) *Every cancellative category that admits a bounded Garside family is an Ore category.*

(ii) *Every bounded Garside family is strong.*

The most important technical property implied by the existence of the bounding map Δ is the existence of a derived automorphism of the ambient category.

Lemma 1.29. [15, Proposition VI.1.11] *Assume that \mathcal{S} is a Garside family bounded by Δ in a cancellative category \mathcal{C} . Put $\phi(x) = \text{trg}(\Delta(x))$ for x in $\text{Obj}(\mathcal{C})$ and $\phi(g) = \partial^2(g)$ for g in \mathcal{C} . Then ϕ is an automorphism of \mathcal{C} that makes the diagram aside commutative for every g in $\mathcal{C}(x, y)$.*



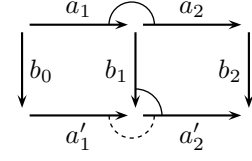
In terms of normal decompositions, the bounding map implies a simple connection between greediness and left-disjointness that need not be valid in general.

Lemma 1.30. [15, Proposition VI.1.46] *Assume that \mathcal{S} is a Garside family bounded by a map Δ in a cancellative category \mathcal{C} . Then, for all a_1, a_2 in \mathcal{S}^\sharp , the following are equivalent:*

- (i) $a_1|a_2$ is \mathcal{S} -normal;
- (ii) $\partial(a_1)$ and a_2 are left-disjoint;
- (iii) $\partial(a_1)$ and a_2 admit no nontrivial common left-divisor.

A consequence is the following counterpart of the first domino rule.

Lemma 1.31. [15, Lemma VI.1.32] (*second domino rule*) Assume that \mathcal{C} is a cancellative category and \mathcal{S} is a bounded Garside family in \mathcal{C} . Then, when we have a commutative diagram as on the right with edges in \mathcal{S}^\sharp in which $a_1|a_2$ and $b_1|a'_2$ are \mathcal{S} -greedy, then $a'_1|a'_2$ is \mathcal{S} -greedy as well.



Note that the second domino rule is not an exact counterpart of the first one in that, here, all involved elements are supposed to lie in the reference family \mathcal{S}^\sharp .

Also, a second type of distinguished decomposition naturally arises for the elements of the associated groupoid of the considered category: called Δ -normal, these decompositions differ from symmetric normal decompositions in that the denominator is demanded to involve a power of the Garside map Δ . In the case of a monoid M , a bound Δ for a Garside family is an element of M , and it makes sense to take powers of Δ . In the case of a general category \mathcal{C} , the bound is a map of $\text{Obj}(\mathcal{C})$ to \mathcal{C} , and the notion of a power has to be adapted.

Notation 1.32. Assume that \mathcal{C} is a cancellative category and Δ is a Garside map in \mathcal{C} . For n in \mathbb{Z} and x in $\text{Obj}(\mathcal{C})$, we put

$$(1.33) \quad \Delta^{[n]}(x) := \begin{cases} \Delta(x)|\Delta(\phi(x))|\cdots|\Delta(\phi^{n-1}(x)) & \text{for } n > 0, \\ \varepsilon_x & \text{for } n = 0, \\ \overline{\Delta(\phi^{-1}(x))|\Delta(\phi^{-2}(x))|\cdots|\Delta(\phi^{-|n|}(x))} & \text{for } n < 0, \end{cases}$$

and we write $\Delta^{(n)}(x)$ for the element of $\text{Env}(\mathcal{C})$ represented by $\Delta^{[n]}(x)$.

Note that, in every case, the source of $\Delta^{[n]}(x)$ and $\Delta^{(n)}(x)$ is x and its target is $\phi^n(x)$, and that $\Delta^{(-n)}(x)$ is always the inverse of $\Delta^{(n)}(\phi^{-n}(x))$.

Definition 1.34. Assume that \mathcal{S} is a Garside family of \mathcal{C} bounded by a map Δ in an Ore category \mathcal{C} .

- (i) An element g of $\mathcal{C}(x, -)$ is called Δ -like if $g =^\times \Delta(x)$ holds.
- (ii) A signed \mathcal{S} -path is called Δ -normal if it has the form $\Delta^{[n]}(-)|a_1|\cdots|a_p$ with n in \mathbb{Z} and $a_1|\cdots|a_p$ an \mathcal{S} -normal path such that a_1 is not Δ -like.

Thus a Δ -normal path is either a positive \mathcal{S} -path beginning with elements in the image of Δ , or an empty path, or a negative–positive \mathcal{S} -path whose negative part consists of elements in the image of Δ .

Proposition 1.35. [15, Proposition VI.3.19] Assume that \mathcal{S} is a Garside family of \mathcal{C} bounded by a map Δ in an Ore category \mathcal{C} . Then every element of $\text{Env}(\mathcal{C})$ admits a Δ -normal decomposition, in which the exponent of Δ is uniquely determined and the other entries are unique up to \mathcal{C}^\times -deformation.

In the positive case, it is easily seen that Δ -like entries must lie at the beginning of an \mathcal{S} -normal path, so a Δ -normal path is simply an \mathcal{S} -normal path in which the initial Δ -like entries are not only Δ -like but even lie in the image of Δ . In the signed case, the difference with a symmetric \mathcal{S} -normal path is more important: in a Δ -normal path, the numerator and the denominator need not be left-disjoint, the requirement being now that Δ does not left-divide the first positive entry. Actually, Propositions 1.21 and 1.35 are not exactly comparable, as they require different assumptions, namely the existence of left-lcms in the former case, and that of a bounded Garside family in the latter.

2. RECOGNIZING GARSIDE FAMILIES, CASE OF A PRESENTATION

We now begin to investigate the effective methods relevant for Garside structures. In this section as well as the next one, we address the question of recognizing that a given family is a Garside family in the ambient category, as well as checking that the category is eligible for the Garside approach, that is, it is left-cancellative (*resp.* left-Ore). The question depends in turn on

the way the category and the candidate Garside family are specified. In this section, we consider the case when the category is specified using a presentation.

We first define reversing (Subsection 2.1), then address establishing that the ambient category is cancellative (Subsection 2.2), recognizing Garside families (Subsection 2.3), and finally proving further properties like being an Ore category or a strong Garside family (Subsection 2.4).

2.1. The reversing transformation. Before entering the main development, we introduce a technical tool that will be used several times in the sequel, namely a path transformation called reversing [6, 9, 12].

Definition 2.1. Assume that \mathcal{A} is a precategory. A *right-complement* on \mathcal{A} is a partial map θ of \mathcal{A}^2 to \mathcal{A}^* such that $\theta(a, a)$ is defined and equal to ε_y for every a in $\mathcal{A}(x, y)$ and that, if $\theta(a, b)$ is defined, then a and b have the same source, $\theta(b, a)$ is defined, and both $a|\theta(a, b)$ and $b|\theta(b, a)$ are defined and have the same target. A right-complement is called *short* if, whenever $\theta(a, b)$ is defined, $\theta(a, b)$ has length at most 1, that is, it belongs to \mathcal{A} or is empty.

Note that, if $\text{Obj}(\mathcal{A})$ consists of a single element, the conditions for θ to be a right-complement are simply that $\theta(a, a)$ is empty for every a and that $\theta(a, b)$ is defined if and only if $\theta(b, a)$ is.

Definition of right-reversing. We are now ready to introduce right-reversing. We begin with the special case of a short right-complement and a negative–positive input path, which will be frequently used in the sequel.

Notation 2.2. If \mathcal{A} is a precategory, we write $\varepsilon_{\mathcal{A}}$ for $\{\varepsilon_x \mid x \in \text{Obj}(\mathcal{A})\}$ and $\widehat{\mathcal{A}}$ for $\mathcal{A} \cup \varepsilon_{\mathcal{A}}$. For θ a right-complement on \mathcal{A} , we write $\widehat{\theta}$ for the extension of θ to $\widehat{\mathcal{A}}$ obtained by adding $\widehat{\theta}(\varepsilon_x, \varepsilon_x) = \varepsilon_x$ for every object x plus $\widehat{\theta}(a, \varepsilon_x) = \varepsilon_y$ and $\widehat{\theta}(\varepsilon_x, a) = a$ for every a in $\mathcal{A}(x, y)$.

Note that $\widehat{\theta}$ is a right-complement on $\widehat{\mathcal{A}}$. We introduce right-reversing by means of an algorithm working on (certain) signed \mathcal{A} -paths.

Algorithm 2.3 (Right-reversing, short case, negative–positive input).

Context: A precategory \mathcal{A} , a short right-complement θ on \mathcal{A}

Input: A negative–positive \mathcal{A} -path $\overline{b_q} \cdots \overline{b_1} | a_1 \cdots | a_p$

Output: A positive–negative \mathcal{A} -path, or **fail**

```

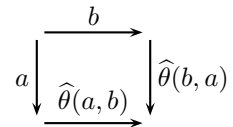
1:  $b_{i,0} := b_i$  for  $i = 1, \dots, q$ 
2:  $a_{0,j} := a_j$  for  $j = 1, \dots, p$ 
3: for  $i$  increasing from 1 to  $q$  do
4:   for  $j$  increasing from 1 to  $p$  do
5:     if  $\widehat{\theta}(a_{i-1,j}, b_{i,j-1})$  is defined then
6:        $b_{i,j} := \widehat{\theta}(a_{i-1,j}, b_{i,j-1})$ 
7:        $a_{i,j} := \widehat{\theta}(b_{i,j-1}, a_{i-1,j})$ 
8:     else
9:       return fail
10: return  $a_{q,1} \cdots | a_{q,p} | \overline{b_{q,p}} \cdots | \overline{b_{1,p}}$ 

```

Lemma 2.4. *If θ is a short right-complement on a finite precategory \mathcal{A} , Algorithm 2.3 running on a pair of \mathcal{A} -paths of length at most ℓ terminates in $O(\ell^2)$ steps.*

Proof. The claim is obvious from the pseudocode in Algorithm 2.3. □

Running Algorithm 2.3 amounts to recursively constructing a rectangular grid whose edges are labeled by elements of \mathcal{A} or empty paths, and in which each elementary square has the type shown aside, see Figure 3 for an example.



We now turn to the case of an arbitrary right-complement, that is, we no longer assume that $\theta(a, b)$ necessarily has length zero or one. Then we can extend the definition of right-reversing,

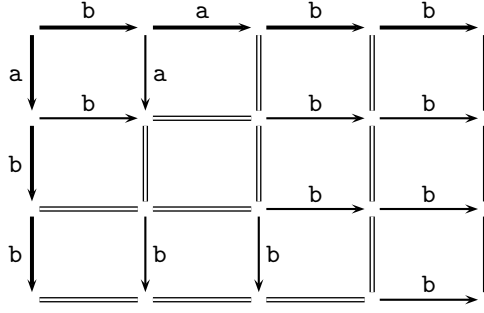


FIGURE 3. The grid associated with a right-reversing, short case. Here we consider the right-complement θ on $\{a, b\}$ defined by $\theta(a, a) = \theta(b, b) = \varepsilon$, $\theta(a, b) = b$, and $\theta(b, a) = a$ (with one object only), and apply Algorithm 2.3 to the negative–positive word $\overline{b|b|a|b|b|b}$: the initial word is written on the left (negative part, here $a|b|b$) and the top (positive part, here $b|a|b|b$), and then the grid is constructed by using the right-complement $\hat{\theta}$ to recursively complete the squares; we use a double line for ε -labeled arrows. In the current case, the output of the algorithm is the length one word b .

the only differences being that the constructed grid may involve rectangles whose edges contain more than one arrow when $\theta(a, b)$ has length at least 2. For the description, it is convenient to start from an arbitrary signed path, and not necessarily from a negative–positive one.

Algorithm 2.5 (Right-reversing, general case).

Context: A precategory \mathcal{A} , a right-complement θ on \mathcal{A}

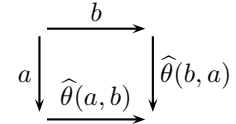
Input: A signed \mathcal{A} -path w

Output: A positive–negative \mathcal{A} -path, or **fail**, or no output

- 1: **while** $\exists i < \|w\|$ ($w[i] \in \overline{\mathcal{A}}$ and $w[i+1] \in \mathcal{A}$) **do**
- 2: $j := \min\{i \mid w[i] \in \overline{\mathcal{A}} \text{ and } w[i+1] \in \mathcal{A}\}$
- 3: $a := \overline{w[j]}$
- 4: $b := w[j+1]$
- 5: **if** $a = b$ **then**
- 6: remove $\overline{a|b}$ in w
- 7: **else**
- 8: **if** $\hat{\theta}(a, b)$ is defined **then**
- 9: replace $\overline{a|b}$ in w with $\hat{\theta}(a, b)|\overline{\hat{\theta}(b, a)}$
- 10: **else**
- 11: **return fail**
- 12: **return** w

Definition 2.6. If Algorithm 2.5 terminates successfully, we say that the initial signed \mathcal{A} -path w is *right- θ -reversible* to the final path w' , and we write $w \curvearrowright_{\theta} w'$. If u, v are positive \mathcal{A} -paths, and there exist positive paths u', v' satisfying $\overline{u|v} \curvearrowright_{\theta} v'|u'$, we define $\theta^*(u, v)$ to be v' and $\theta^*(v, u)$ to be u' .

As in the short case, running Algorithm 2.5 amounts to recursively constructing a grid whose edges are labeled by elements of \mathcal{A} or empty paths, and in which each elementary square has the type shown aside. The difference to Algorithm 2.3 is twofold:



Firstly, if the input is an arbitrary signed path, we do not necessary start with a vertical–horizontal path, but possibly with a staircase in which vertical and horizontal edges alternate; see Figure 4 for an example.

Secondly, if the right-complement θ is not short, the edges of the grid may have different sizes; see Figure 5 for an example.

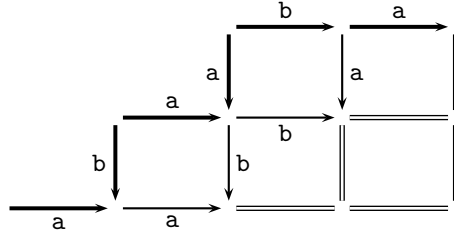


FIGURE 4. The grid associated with the right-reversing of a signed path. We consider the right-complement from Figure 3, and apply Algorithm 2.5 to the word $a|\bar{b}|a|\bar{a}|b|a$: the initial word is written on the top-left boundary (negative edges vertically and positive edges horizontally), and completing the grid to the bottom-right yields the output word $a|a$.

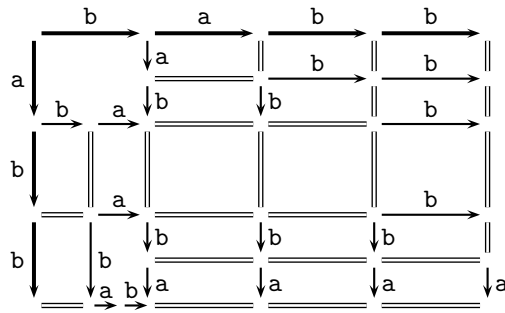


FIGURE 5. The grid associated with a right-reversing for a right-complement that is not short. Here we consider the right-complement θ on $\{a, b\}$ defined by $\theta(a, a) = \theta(b, b) = \varepsilon$, $\theta(a, b) = b|a$, and $\theta(b, a) = a|b$ (with one object only), and apply Algorithm 2.5 to the negative-positive word $\bar{b}|\bar{b}|\bar{a}|b|a|b|b$ (the same as in Figure 3): the difference is that, now, edges of variable size occur, so that, a priori, the process need not terminate. In the current case, it terminates, and the output word is $a|b|\bar{a}$.

Remark 2.7. More general versions of right-reversing are possible: at the expense of renouncing to determinism, we can consider multiform right-complements assigning with every pair of letters (a, b) a family of pairs of paths $\{(u_1, v_1), \dots, (u_n, v_n)\}$ and decide that $\bar{a}|b$ may reverse to any of the paths $v_1|\bar{u}_1, \dots, v_n|\bar{u}_n$, see [9]. In such a context, several reversing grids may be associated with an initial path. Although most theoretical results can be adapted, these extended versions are less suitable for algorithms, and we shall not consider them here.

Termination of reversing. It should be clear that, whereas Algorithm 2.3 always terminates (successfully or not, that is, with an output path or with the output “fail”) in finitely many steps, Algorithm 2.5 may not terminate.

Example 2.8. Consider the right-complement θ defined on $\{a, b\}$ by $\theta(a, b) = b$ and $\theta(b, a) = b|a$. Let $w = \bar{a}|b|a$. Then w reverses in two steps to $b|w|\bar{b}$, hence in $2n$ steps to $b^n|w|\bar{b}^n$ for every n , never leading to a positive-negative word.

For our current approach, it will be useful to have a simple termination criterion.

Lemma 2.9. *Assume that θ is a right-complement on a precategory \mathcal{A} .*

(i) *Right- θ -reversing terminates successfully for all valid inputs if and only if there exists a family \mathcal{B} of \mathcal{A} -paths that includes \mathcal{A} and is such that, for all u, v in \mathcal{B} with the same source, there exist u', v' in \mathcal{B} satisfying $\overline{u|v} \sim_{\theta} v'|\overline{u'}$.*

(ii) *If a family \mathcal{B} with the properties as in (i) exists and is finite, then, for every signed \mathcal{A} -path of length ℓ , the right- θ -reversing of w terminates (successfully) in $O(\ell^2)$ steps and all involved paths have length in $O(\ell)$.*

Proof. (i) If right- θ -reversing is terminating, then \mathcal{A}^* has the expected property.

Conversely, assume that \mathcal{B} satisfies the property of the lemma. Let θ' be the restriction of θ^* to \mathcal{B} . Then θ' is a short right-complement on \mathcal{B} , so right- θ' -reversing is terminating. Now, assume that $\overline{u|v}$ is a signed \mathcal{A} -path. By assumption, $\overline{u|v}$ is also a signed \mathcal{B} -path, and its right- θ' -reversing terminates, so there exists a witnessing θ' -grid. Now a θ' -grid of size $p \times q$ is the juxtaposition of pq θ -grids, whose existence shows that the right- θ -reversing of $\overline{u|v}$ also terminates.

(ii) Right- θ' -reversing terminates in $O(\ell^2)$ steps and all involved \mathcal{B} -paths have length in $O(\ell^2)$. As the family \mathcal{B} is finite, there exists a constant that bounds the length of any element of \mathcal{B} considered as an \mathcal{A} -path, and the claim follows. \square

Corollary 2.10. *If θ is a short right-complement on a finite precategory \mathcal{A} , Algorithm 2.5 running on a signed \mathcal{A} -path of length ℓ terminates in $O(\ell^2)$ steps.*

Proof. If θ is short, the condition of Lemma 2.9 is satisfied with \mathcal{B} equal to the family $\widehat{\mathcal{A}}$ in Notation 2.2, that is, the union of \mathcal{A} and the empty paths. \square

So, starting with an arbitrary right-complement θ , we can possibly show that right- θ -reversing terminates successfully for all valid inputs by applying the following closure method:

Algorithm 2.11 (Termination of right-reversing).

Context: A precategory \mathcal{A}

Input: A right-complement θ on \mathcal{A}

Output: A (minimal) subfamily of \mathcal{A}^* that includes \mathcal{A} and is closed under θ^*

- 1: $\mathcal{A}_0 := \mathcal{A}$
- 2: **repeat**
- 3: $\mathcal{A}_{i+1} := \mathcal{A}_i \cup \{\theta^*(a, b) \mid a, b \in \mathcal{A}_i\}$
- 4: **until** $\mathcal{A}_{i+1} = \mathcal{A}_i$
- 5: **return** \mathcal{A}_i

Example 2.12. Consider the right-complement of Figure 5 again. Starting with $\mathcal{A}_0 = \{\mathbf{a}, \mathbf{b}\}$ and applying Algorithm 2.11, we find $\mathcal{A}_1 = \mathcal{A}_0 \cup \{\varepsilon, \mathbf{a|b}, \mathbf{b|a}\}$, and $\mathcal{A}_2 = \mathcal{A}_1$: here the process terminates in one step, leading to $\{\varepsilon, \mathbf{a}, \mathbf{b}, \mathbf{a|b}, \mathbf{b|a}\}$. The existence of this 5 element family that is closed under right-reversing implies that right-reversing is terminating with a quadratic complexity upper bound.

When the closure under θ^* is infinite, the situation is more complicated and there is no general result. Examples are known when right-reversing is always terminating but the time (*resp.* space) complexity is more than quadratic (*resp.* linear): for instance, the right-reversing associated with the right-complement θ defined on $\{\mathbf{a}, \mathbf{b}\}$ by $\theta(\mathbf{a}, \mathbf{b}) = \varepsilon$ and $\theta(\mathbf{b}, \mathbf{a}) = \mathbf{a}(\mathbf{bab})^3\mathbf{ab}$ is always terminating, but the time (*resp.* space) complexity is cubic (*resp.* quadratic) [13, Example 10.3], whereas [7, Proposition VIII.1.11] displays an example (with an infinite family of generators) where right-reversing is terminating but the only known bound for time complexity is a tower of exponentials of exponential height.

2.2. Establishing left-cancellativity. We now address our main problem, namely investigating a presented category $\langle \mathcal{A} \mid \mathcal{R} \rangle^+$ and, in particular, trying to recognize whether it is left-cancellative. Here we consider the problem for presentations of a certain syntactical type. This restriction allows for using right-reversing, and it is natural in our context as one can show that every

Garside family gives rise to a presentation that is eligible for this approach (at least in the extended version alluded to in Remark 2.7).

Definition 2.13. A presentation $(\mathcal{A}; \mathcal{R})$ is called *positive* if all relations of \mathcal{R} are of the form $u = v$ with u, v nonempty; it is called *right-complemented*, associated with the right-complement θ , if \mathcal{R} consists of all relations $a|\theta(a, b) = b|\theta(b, a)$ with (a, b) in the domain of θ .

Note that, by definition, a right-complemented presentation is positive and that, if $(\mathcal{A}; \mathcal{R})$ is a positive presentation, then the category $\langle \mathcal{A} | \mathcal{R} \rangle^+$ contains no nontrivial invertible element since an empty path and a nonempty path cannot be \mathcal{R} -equivalent. Saying that a presentation $(\mathcal{A}; \mathcal{R})$ is right-complemented just means that it is positive and that, for all a, b in \mathcal{A} , the family \mathcal{R} contains at most one relation of the form $a\dots = b\dots$. The involved right-complement is short if the paths “...” have length 0 or 1, that is, all relations in \mathcal{R} are of the form $u = v$ with u and v of length 1 or 2. For instance, the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{ab} = \mathbf{ba})$ is associated with the short right-complement of Figure 3, whereas the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{aba} = \mathbf{bab})$ is associated with the right-complement of Figure 5 and Example 2.12. By contrast, the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{ab} = \mathbf{ba}, \mathbf{a}^2 = \mathbf{b}^2)$ is not right-complemented since it contains two relations of the form $\mathbf{a}\dots = \mathbf{b}\dots$.

When a presentation is right-complemented, it is eligible for the associated right-reversing transformation, leading in good cases to a practical method for recognizing left-cancellativity. The first observation is that right-reversing gives a way to construct common right-multiples in the associated category:

Lemma 2.14. *Assume that $(\mathcal{A}; \mathcal{R})$ is a presentation associated with a right-complement θ . Then, for all \mathcal{A} -paths u, v, u', v' satisfying $\bar{u}|v \curvearrowright_{\theta} v'|\bar{u}'$, the paths $u|v'$ and $v|u'$ are \mathcal{R} -equivalent.*

Proof. By definition, each elementary square in the rectangular grid associated with Algorithm 2.5 corresponds to a relation of \mathcal{R} . \square

Lemma 2.14 says in particular that, if u, v are \mathcal{A} -paths and $\bar{u}|v$ is right- θ -reversible to an empty path, then u and v are \mathcal{R} -equivalent, that is, they represent the same element in the category $\langle \mathcal{A} | \mathcal{R} \rangle^+$. In our context, right-reversing will be useful only when the previous implication is an equivalence.

Definition 2.15. If $(\mathcal{A}; \mathcal{R})$ is a presentation associated with a right-complement θ , we say that right-reversing is *complete* for $(\mathcal{A}; \mathcal{R})$ if $\bar{u}|v \curvearrowright_{\theta} \varepsilon$ holds whenever u and v represent the same element in $\langle \mathcal{A} | \mathcal{R} \rangle^+$.

In other words, right-reversing is complete if it always detects equivalence. The interest of introducing completeness here is the following easy result:

Lemma 2.16. *Assume that $(\mathcal{A}; \mathcal{R})$ is a presentation associated with a right-complement θ and right-reversing is complete for $(\mathcal{A}; \mathcal{R})$. Then the category $\langle \mathcal{A} | \mathcal{R} \rangle^+$ is left-cancellative.*

Proof. It is sufficient to prove that, if a belongs to \mathcal{A} and u, v are \mathcal{A} -paths such that $a|u$ and $a|v$ are \mathcal{R} -equivalent, then u and v are \mathcal{R} -equivalent. Now, as right-reversing is complete, the hypothesis implies that $\bar{u}|\bar{a}|a|v$ right- θ -reverses to an empty path. Now the first step in the reversing process necessarily consists in deleting $\bar{a}|a$. We deduce that $\bar{u}|v$ must right- θ -reverse to an empty path, which, by Lemma 2.14, implies that u and v are \mathcal{R} -equivalent. \square

We are thus led to looking for completeness criteria.

Definition 2.17. Assume that θ is a right-complement on a precategory \mathcal{A} . For a, b, c in \mathcal{A} , we say that θ satisfies the *cube condition* at (a, b, c) if either neither of $\theta^*(\theta^*(a, b), \theta^*(a, c))$ and $\theta^*(\theta^*(b, a), \theta^*(b, c))$ is defined, or both are defined and

$$\theta^*(\theta^*(\theta^*(a, b), \theta^*(a, c)), \theta^*(\theta^*(b, a), \theta^*(b, c)))$$

is empty.

Proposition 2.18. [15, Proposition II.4.11] or [8, Proposition 6.1] *Assume that $(\mathcal{A}; \mathcal{R})$ is a presentation associated with a right-complement θ and, moreover, the right-complement θ is short, or the presentation $(\mathcal{A}; \mathcal{R})$ is right-Noetherian. Then right-reversing is complete for $(\mathcal{A}; \mathcal{R})$ if and only if, for all pairwise distinct a, b, c in \mathcal{A} with the same source, the cube condition for (a, b, c) is satisfied.*

Corollary 2.19. *If the equivalent conditions of Proposition 2.18 are satisfied, the category $\langle \mathcal{A} | \mathcal{R} \rangle^+$ is left-cancellative.*

Example 2.20. Both presentations $(\mathbf{a}, \mathbf{b}; \mathbf{ab} = \mathbf{ba})$ and $(\mathbf{a}, \mathbf{b}; \mathbf{aba} = \mathbf{bab})$ are eligible for Proposition 2.18 and Corollary 2.19. Indeed, the former is associated with a short right-complement, whereas the latter is associated with a right-complement that is not short, but defining $\lambda(w)$ to be the length of w and noting that the (unique) relation of the presentation consists of two words with the same length, Lemma 1.6 yields that the presentation is right-Noetherian. Then, in order to apply Proposition 2.18 and Corollary 2.19, we need to check that the cube condition is satisfied for all triples of pairwise distinct elements of $\{\mathbf{a}, \mathbf{b}\}$, which is vacuously true. We conclude that both monoids $\langle \mathbf{a}, \mathbf{b} | \mathbf{ab} = \mathbf{ba} \rangle^+$ (the free Abelian monoid \mathbb{N}^2) and $\langle \mathbf{a}, \mathbf{b} | \mathbf{aba} = \mathbf{bab} \rangle^+$ (the 3-strand braid monoid B_3^+) are left-cancellative.

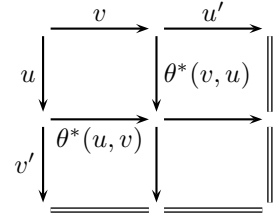
Remark 2.21. Proposition 2.18 does not exhaust all known types of presentations for which right-reversing is complete. For instance, the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{a} = \mathbf{bab})$ of the Klein bottle monoid is not eligible for Proposition 2.18, since the associated right-complement θ is neither short nor right-Noetherian since no map λ may satisfy $\lambda(\mathbf{a}) = \lambda(\mathbf{bab}) > \lambda(\mathbf{ab})$. However right-reversing is complete and terminating for this presentation, as well as for a number of similar presentations [13].

2.3. Recognizing Garside. Assuming that $(\mathcal{A}; \mathcal{R})$ is a right-complemented presentation and the category $\langle \mathcal{A} | \mathcal{R} \rangle^+$ has been shown to be left-cancellative using the method explained in Subsection 2.2, our next task is to recognize that some subfamily is possibly a Garside family. Once again the task will turn out to be easy whenever right-reversing is complete for the considered presentation.

The main observation is that, in the above context, the category $\langle \mathcal{A} | \mathcal{R} \rangle^+$ admits local right-lcms, that is, any two elements that admit a common right-multiple admit a right-lcm.

Lemma 2.22. *Assume that $(\mathcal{A}; \mathcal{R})$ is a presentation associated with a right-complement θ and right-reversing is complete for it. Then, for all \mathcal{A} -paths u, v with the same source, the elements $[u], [v]$ admit a common right-multiple in $\langle \mathcal{A} | \mathcal{R} \rangle^+$ if and only if the right-reversing of $\bar{u}|v$ terminates, in which case we have $[u] \vee [v] = [u|\theta^*(u, v)]$ and $[u] \setminus [v] = [\theta^*(u, v)]$.*

Proof. Assume that $[u]$ and $[v]$ admit a common right-multiple in the category $\langle \mathcal{A} | \mathcal{R} \rangle^+$. This means that there exist \mathcal{A} -paths u', v' such that $u|v'$ and $v|u'$ are \mathcal{R} -equivalent. As right-reversing is complete, this implies that $\bar{v}'|\bar{u}|v|u'$ right-reverses to an empty path. Decompose the associated reversing grid as shown aside. The assumption that the reversing of $\bar{v}'|\bar{u}|v|u'$ successfully terminates implies in particular that the reversing of $\bar{u}|v$ successfully terminates, that is, the paths $\theta^*(u, v)$ and $\theta^*(v, u)$ are defined.



The diagram then shows that $[u|v']$ is a right-multiple of $[u|\theta^*(u, v)]$ in $\langle \mathcal{A} | \mathcal{R} \rangle^+$, so the latter, which depends only on $[u]$ and $[v]$, is a right-lcm of these elements. \square

Applying the criterion of Lemma 1.13, we immediately deduce a method for recognizing Garside families in the right-Noetherian case. Actually, we obtain more: instead of just a YES/NO answer for a candidate-subfamily, we obtain the existence and a characterization of the smallest Garside family that includes the given family. Hereafter, for $\mathcal{B} \subseteq \mathcal{A}^*$, we write $[\mathcal{B}]$ for $\{[w] \mid w \in \mathcal{B}\}$.

Algorithm 2.23 (Smallest Garside family).

Context: A right-Noetherian right-complemented presentation $(\mathcal{A}; \mathcal{R})$ for which right-reversing is complete

Input: A finite subfamily \mathcal{B} of \mathcal{A}^* that includes \mathcal{A}

Output: A subfamily of \mathcal{A}^* that represents the smallest Garside family of $\langle \mathcal{A} \mid \mathcal{R} \rangle^+$ that includes $\mathcal{B} \cup \mathbf{1}_{\mathcal{A}}$ if the algorithm terminates successfully

```

1: enumerate  $\mathcal{B}$  as  $\{w_1, \dots, w_n\}$  and set  $\widehat{\mathcal{B}} := [w_1, \dots, w_n]$ 
2:  $i := 2$ 
3: while  $i \leq |\widehat{\mathcal{B}}|$  do
4:   for  $j := 1$  to  $i - 1$  do
5:     if  $\theta^*(w_i, w_j)$  is defined then                                 $\triangleright$  right-reversing may fail to terminate
6:       INCLUDE( $\widehat{\mathcal{B}}, \theta^*(w_i, w_j)$ )
7:       INCLUDE( $\widehat{\mathcal{B}}, \theta^*(w_j, w_i)$ )
8:       INCLUDE( $\widehat{\mathcal{B}}, w_i | \theta^*(w_i, w_j)$ )
9:       INCLUDE( $\widehat{\mathcal{B}}, w_j | \theta^*(w_j, w_i)$ )
10:    else
11:      return fail
12:     $i := i + 1$ 
13: return  $\widehat{\mathcal{B}}$ 
14: procedure INCLUDE( $\widehat{\mathcal{B}}, w$ )       $\triangleright$  append  $w$  to  $\widehat{\mathcal{B}}$ , unless  $\widehat{\mathcal{B}}$  contains a path equivalent to  $w$ .
15:   if  $\nexists i \in \{1, \dots, |\widehat{\mathcal{B}}|\}$  with  $\theta^*(w, w_i) = \theta^*(w_i, w) = \varepsilon_-$  then
16:     append  $w$  to  $\widehat{\mathcal{B}}$ 

```

Proposition 2.24. *Assume that $(\mathcal{A}; \mathcal{R})$ is a finite right-Noetherian right-complemented presentation for which right-reversing is complete and \mathcal{B} is a finite subfamily of \mathcal{A}^* that includes \mathcal{A} . Then there exists a smallest finite Garside family of \mathcal{C} including $[\mathcal{B}] \cup \mathbf{1}_{\mathcal{C}}$ if and only if Algorithm 2.23 successfully terminates, in which case the returned family of paths represents that Garside family.*

Proof. By Lemma 2.22, any two elements of $\langle \mathcal{A} \mid \mathcal{R} \rangle^+$ that admit a common right-multiple admit a right-lcm, so $\langle \mathcal{A} \mid \mathcal{R} \rangle^+$ is eligible for Lemma 1.13. Hence a subfamily \mathcal{S} that includes $\mathcal{A} \cup \mathbf{1}_{\mathcal{A}}$ is a Garside family if and only if, for all a, b in \mathcal{S} with a common right-multiple, $a \vee b$ and $a \setminus b$ belong to \mathcal{S} . So, by Lemma 2.22 again, a family $\widehat{\mathcal{B}}$ of \mathcal{A} -paths represents a Garside family if and only if, for all u, v in $\widehat{\mathcal{B}}$, the paths $u | \theta^*(u, v)$ and $\theta^*(u, v)$ are \mathcal{R} -equivalent to at least one element of $\widehat{\mathcal{B}}$. It follows that there exists a smallest Garside family that includes $[\mathcal{B}]$, namely the smallest family of words that includes \mathcal{B} and is such that, for all u, v in $\widehat{\mathcal{B}}$, the paths $u | \theta^*(u, v)$ and $\theta^*(u, v)$ are \mathcal{R} -equivalent to elements of $\widehat{\mathcal{B}}$.

That smallest subfamily $\widehat{\mathcal{B}}$ is precisely what Algorithm 2.23 computes. Indeed, what the latter does is to consider systematically all pairs (w_i, w_j) and, for each of them, test whether $\theta^*(w_i, w_j)$ (lines 6 and 7) as well as $w_i | \theta^*(w_i, w_j)$ (lines 8 and 9) are \mathcal{R} -equivalent to some existing path w_i of the list and, if not, append the missing paths to the list. Note that, as right-reversing is complete, two paths w, w' are equivalent if and only if $\theta^*(w, w')$ and $\theta^*(w', w)$ exist and are empty (line 15). \square

Example 2.25. Consider again the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{aba} = \mathbf{bab})$. Running Algorithm 2.23 on the family $\{\mathbf{a}, \mathbf{b}\}$ yields $\{\varepsilon, \mathbf{a}, \mathbf{b}, \mathbf{a|b}, \mathbf{b|a}, \mathbf{a|b|a}\}$, a family of words representing the well known smallest Garside family $\{1, \mathbf{a}, \mathbf{b}, \mathbf{ab}, \mathbf{ba}, \Delta\}$ of the braid monoid B_3^+ that includes 1; here and everywhere in the sequel, we use Δ for \mathbf{aba} .

Remark 2.26. If right-reversing is not always terminating, that is, if some elements of the considered category have no common right-multiple although they share the same source, Algorithm 2.23 may never terminate. Even in the case when right-reversing is always terminating, it might happen that Algorithm 2.23 does not terminate: for instance, in the case of $(\mathbf{a}, \mathbf{b}; \mathbf{ab}^2 = \mathbf{ba})$, starting from $\{\mathbf{a}, \mathbf{b}\}$, even Algorithm 2.23 does not terminate in finite time: indeed, we have then

$\theta^*(\mathbf{a}, \mathbf{b}^n) = \mathbf{b}^{2n}$, and the family $\widehat{\{\mathbf{a}, \mathbf{b}\}}$ is the infinite family $\{\varepsilon, \mathbf{a}\} \cup \{\mathbf{b}^{2^n} \mid n \geq 0\}$. However, it can be shown [8, Lemme 1.9] that, when the closure under \setminus is finite, then the closure under \setminus and \vee is necessarily finite.

2.4. Further questions. In the previous subsections, we showed how to establish left-cancellativity and to recognize Garside families starting from a presentation. We now briefly address further relevant questions, namely recognizing Ore categories, establishing the existence of lcms, and recognizing strong and bounded Garside families.

Establishing right-cancellativity. Right-reversing is not suitable here, and no practical method is known for establishing right-cancellativity directly. However an obvious solution is to consider the opposite category and the opposite presentation, that is, to switch left and right everywhere, and apply the previous methods. Equivalently, we can work with the initial presentation and apply the symmetric counterpart of right-reversing, naturally called *left-reversing*: whereas right-reversing consists in iteratively replacing subpaths of the form $\overline{a|b}$ with $\theta(a, b)|\overline{\theta(b, a)}$, left-reversing consists in replacing subpaths of the form $b|\overline{a}$ with $\overline{\tilde{\theta}(a, b)}|\tilde{\theta}(b, a)$ when $\tilde{\theta}$ is a *left-complement* on the considered precategory \mathcal{A} , namely a partial map of \mathcal{A}^2 to \mathcal{A}^* such that $\tilde{\theta}(a, a)$ is empty for every a in \mathcal{A} and that, if $\tilde{\theta}(a, b)$ is defined, then a and b have the same target, $\tilde{\theta}(b, a)$ is defined, and both $\tilde{\theta}(a, b)|b$ and $\tilde{\theta}(b, a)|a$ are defined and have the same source. In terms of diagrams, this corresponds to constructing a grid starting from the bottom and the right, instead of from the top and the left. Then the counterpart of Proposition 2.19 gives a criterion for establishing that a category $\langle \mathcal{A} \mid \mathcal{R} \rangle^+$ is right-cancellative.

Establishing the existence of common multiples. Here two different methods can be used. If $(\mathcal{A}; \mathcal{R})$ is a presentation associated with a right-complement θ , then the existence of common right-multiples in $\langle \mathcal{A} \mid \mathcal{R} \rangle^+$ is directly connected with the termination of right-reversing since, as proved in Lemma 2.22, two elements $[u]$ and $[v]$ admit a common right-multiple (and even a right-lcm) in $\langle \mathcal{A} \mid \mathcal{R} \rangle^+$ if and only if the right- θ -reversing of $\overline{u|v}$ successfully terminates in finite time. We deduce the following sufficient condition:

Proposition 2.27. *If $(\mathcal{A}; \mathcal{R})$ is a presentation associated with a right-complement θ and Algorithm 2.11 running on $(\mathcal{A}; \mathcal{R})$ succeeds, any two elements of $\langle \mathcal{A} \mid \mathcal{R} \rangle^+$ with the same source admit a common right-multiple.*

Another approach can be used once a Garside family \mathcal{S} is known. It is based on the following result, which reduces the existence of common multiples for arbitrary elements to the existence of common multiples inside the Garside family.

Proposition 2.28. *Assume that \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} . Then any two elements of \mathcal{C} with the same source admit a common right-multiple if and only if any two elements of \mathcal{S} with the same source admit one.*

Proof. Obviously the condition is necessary. On the other hand, assume that any two elements of \mathcal{S} with the same source admit a common right-multiple. By Lemma 1.14, for all a, b in \mathcal{S}^\sharp with the same source, there exist a', b' in \mathcal{S}^\sharp satisfying $ab' = ba'$. Now, consider arbitrary elements f, g of \mathcal{C} with the same source. As \mathcal{S}^\sharp generates \mathcal{C} , there exist a_1, \dots, a_p and b_1, \dots, b_q in \mathcal{S}^\sharp satisfying $f = a_1 \cdots a_p$ and $g = b_1 \cdots b_q$. Using the result above, one inductively constructs a $p \times q$ rectangular grid based on a_1, \dots, a_p and b_1, \dots, b_q with edges in \mathcal{S}^\sharp , and the diagonal of the grid (as well as any path from the top-left corner to the bottom-right corner) represents a common right-multiple of f and g . \square

Note that the construction of a grid in the proof of Proposition 2.28 is directly reminiscent of a right-reversing process—more exactly, of its non-deterministic extension alluded to in Remark 2.7 as there is no uniqueness of the elements called a', b' in general.

It should be clear that Proposition 2.28 directly leads to an effective method for deciding the existence of common right-multiples in the case of a finite Garside family, see Example 2.31 below.

A symmetric argument is possible for common left-multiples. However, as the definition of a Garside family is not invariant under exchanging left and right, the result takes a different form. In particular, it only gives a sufficient condition that need not be necessary in general.

Proposition 2.29. *Assume \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} and, for all a, b in \mathcal{S}^\sharp with the same target, there exist a', b' in \mathcal{S}^\sharp satisfying $a'b = b'a$. Then any two elements of \mathcal{C} with the same target admit a common left-multiple.*

The proof is symmetric to that of Proposition 2.28, the inductive step consisting now in constructing a rectangular grid starting from the bottom and the right. Again Proposition 2.29 leads to an effective method for deciding the existence of common left-multiples in the case of a finite Garside family, see Example 2.31.

Establishing the existence of lcms. As for the existence of right-lcms (their computation will be addressed in Subsection 4.4), we shall just mention an algorithmically important consequence of Lemma 1.14:

Proposition 2.30. *Assume that \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} . Then \mathcal{C} admits right-lcms (resp. local right-lcms) if and only if any two elements a, b of \mathcal{S}^\sharp with the same source (resp. that admit a common right-multiple in \mathcal{S}^\sharp) admit a right-lcm c inside \mathcal{S}^\sharp , this meaning that c belongs to \mathcal{S}^\sharp and every common right-multiple of a and b that lies in \mathcal{S}^\sharp is a right-multiple of c .*

Proof. By Lemma 1.14, a common right-multiple of two elements of \mathcal{S}^\sharp must be a right-multiple of some common right-multiple that lies in \mathcal{S}^\sharp . So it is enough to consider right-multiples lying in \mathcal{S}^\sharp , and we deduce that two elements of \mathcal{S}^\sharp admit a right-lcm in \mathcal{C} if and only if they admit one inside \mathcal{S}^\sharp . Then (the proof of) Proposition 2.28 enables one to go from \mathcal{S}^\sharp to products of elements of \mathcal{S}^\sharp , that is, to arbitrary elements of \mathcal{C} . \square

We can thus establish the possible existence of right-lcms by exclusively inspecting right-multiples inside the considered Garside family.

Recognizing strong and bounded Garside families. So far, we have obtained methods for possibly establishing that a category is left- or right-Ore. Deciding whether a given Garside family \mathcal{S} is strong is then easy when \mathcal{S} is finite and an effective method is available for deciding, for all f, g, h , whether h is a left-lcm of f and g . Note that, by the counterpart of Lemma 2.22, such a method exists whenever the presentation is associated with a left-complement for which left-reversing is complete. Finally, deciding whether a finite Garside family is bounded is easy, as it only requires testing divisibility relations.

Example 2.31. We continue with the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{aba} = \mathbf{bab})$ of the monoid B_3^+ . In Example 2.25, we identified the Garside family $\mathcal{S} = \{1, \mathbf{a}, \mathbf{b}, \mathbf{ab}, \mathbf{ba}, \Delta\}$ (where we recall Δ is \mathbf{aba}). It is clear that Δ is a right-multiple of every element of \mathcal{S} . By Proposition 2.28, we conclude that any two elements of B_3^+ admit a common right-multiple. Next, we easily check the existence of right-lcms inside \mathcal{S} and, by Proposition 2.30, conclude that B_3^+ admits right-lcms (alternatively one can invoke Lemma 2.22 here as the monoid contains no nontrivial invertible element).

Similarly, Δ is also a common left-multiple of each element of \mathcal{S}^\sharp , which is \mathcal{S} , so, by Proposition 2.29, we deduce that any two elements of B_3^+ admit a common left-multiple—whence a left-lcm by the symmetric counterpart of Lemma 2.22.

Finally, as every element of \mathcal{S} is both a left- and a right-divisor of Δ , we deduce that the Garside family \mathcal{S} is bounded by Δ . So, in particular, it is strong.

3. RECOGNIZING GARSIDE FAMILIES, CASE OF A GERM

We now consider the same questions as in Section 2, namely establishing left-cancellativity and recognizing Garside families, when the ambient category is specified by giving either the complete multiplication table (Subsection 3.1), or a germ, defined to be a fragment of the multiplication table that contains enough information to determine the latter unambiguously (Subsections 3.2

and 3.3). In this case, one obtains a compact method that enables one to treat both questions (left-cancellativity and Garside family) simultaneously.

3.1. Using the complete multiplication table. We first quickly consider the case of a finite category. Such a category can be specified by an exhaustive enumeration of its elements, its source and map functions, and its multiplication table, which all are finite data. All questions are then easy.

First, for a finite category with an explicit multiplication table, left-cancellativity can be decided by an exhaustive inspection. Next, the left-divisibility relation is finite, so checking the condition (1.9) is easy and, for every subfamily \mathcal{S} of \mathcal{C} , one can construct a list of all \mathcal{S} -greedy paths of length two. *A priori*, it is not clear that this is sufficient to recognize a Garside family, since the definition of the latter mentions no upper bound on the length of the considered decompositions. However, such a bound exists, which limits both the elements to be considered and the length of the candidate-decompositions.

Lemma 3.1. [14, Proposition 3.1] *A subfamily \mathcal{S} of a left-cancellative category \mathcal{C} is a Garside family if and only if \mathcal{S}^\sharp generates \mathcal{C} and every element of $(\mathcal{S}^\sharp)^2$ admits an \mathcal{S} -normal decomposition of length two.*

We deduce

Proposition 3.2. *Assume that \mathcal{C} is a finite category \mathcal{C} with n elements, and that the source and target maps on \mathcal{C} and the multiplication table of \mathcal{C} are given. Then the following hold:*

- (i) *It is decidable in time $O(n^2)$ whether \mathcal{C} is left-cancellative.*
- (ii) *The left-divisibility relation can be computed in time $O(n^2)$.*
- (iii) *If \mathcal{C} is left-cancellative and \mathcal{S} is a subfamily of \mathcal{C} , then it is decidable in time $O(n^6)$ whether \mathcal{S} is a Garside family in \mathcal{C} .*

Proof. We can test whether \mathcal{C} is left-cancellative and compute the left-divisibility relation as follows: For each f in \mathcal{C} , iterate over all g in \mathcal{C} , keeping track of the products fg that occur; \mathcal{C} is left-cancellative if and only if no such product occurs more than once, and the occurring products are precisely the right-multiples of f . Hence (i) and (ii) hold.

Comparing fg to 1_x , where x is the source of f for all f, g in \mathcal{C} is sufficient to determine \mathcal{C}^\times , and then \mathcal{S}^\sharp can be obtained by computing all products ag for a in \mathcal{S} and g in \mathcal{C}^\times . Thus computing \mathcal{S}^\sharp takes time $O(n^2)$. To verify that \mathcal{S}^\sharp generates \mathcal{C} , one computes, for $i = 2, 3, \dots$, the set $(\mathcal{S}^\sharp)^i$ by considering all products ga with g in $(\mathcal{S}^\sharp)^{i-1}$ and a in \mathcal{S}^\sharp until this sequence stabilises, which happens after at most n steps. Thus verifying that \mathcal{S}^\sharp generates \mathcal{C} takes time $O(n^3)$. Finally, by Lemma 3.1, one can decide whether a subfamily \mathcal{S} of \mathcal{C} is a Garside family by checking for each element of $(\mathcal{S}^\sharp)^2$ all possible \mathcal{S} -paths of length two against condition (1.9); as each test of condition (1.9) involves $O(n^2)$ operations, this takes time $O(n^6)$, completing the proof of (iii). \square

Note that, if \mathcal{C} is infinite, \mathcal{S} being finite does not make recognizing \mathcal{S} -greediness decidable in general, as (1.9) contains a universal quantification over an arbitrary element f of \mathcal{C} and, so, in that case, Lemma 3.1 is of no use.

Remark 3.3. The above analysis partly extends to the case when the category \mathcal{C} is infinite but all operations are computable. For instance, if multiplication, viewed as a partial function of \mathcal{C}^2 to \mathcal{C} , is computable, then left-cancellativity is a Π_1^1 condition: by enumerating all triples (f, g, g') with $g \neq g'$ and checking whether fg and fg' are equal, one finds a counter-example in finite time if one exists, but one never obtains an answer when the category is left-cancellative. Similarly, left-divisibility is a Σ_1^1 condition and, therefore, \mathcal{S} -greediness, which entails an additional existential quantification over an arbitrary element f of \mathcal{C} , is a Π_2^1 condition.

3.2. Germs. For an infinite category, it is impossible to exhaustively enumerate the multiplication table. However, in good cases, it may happen that some finite fragment of the latter determines the category and provides methods for establishing properties of the latter. This is the germ approach that we now introduce.

Definition 3.4. A *germ* is a triple $(\mathcal{S}, \mathbf{1}_{\mathcal{S}}, \bullet)$ where \mathcal{S} is a precategory, $\mathbf{1}_{\mathcal{S}}$ is a subfamily of \mathcal{S} consisting of an element 1_x with source and target x for each object x , and \bullet is a partial map of $\mathcal{S}^{[2]}$ into \mathcal{S} that satisfies

$$(3.5) \quad \text{if } a \bullet b \text{ is defined, its source is the source of } a \text{ and its target is the target of } b,$$

$$(3.6) \quad 1_x \bullet a = a = a \bullet 1_y \text{ hold for each } a \text{ in } \mathcal{S}(x, y),$$

$$(3.7) \quad \text{if } a \bullet b \text{ and } b \bullet c \text{ are defined, then } (a \bullet b) \bullet c \text{ is defined if and only if } a \bullet (b \bullet c) \text{ is, in which case they are equal.}$$

The germ is called *left-associative* if, for all c, a, b in \mathcal{S} , it satisfies

$$(3.8) \quad \text{if } (a \bullet b) \bullet c \text{ is defined, then } b \bullet c \text{ is defined,}$$

and it is called *left-cancellative* if, for all a, b, b' in \mathcal{S} , it satisfies

$$(3.9) \quad \text{if } a \bullet b \text{ and } a \bullet b' \text{ are defined and equal, then } b = b' \text{ holds.}$$

We shall usually write $\underline{\mathcal{S}}$ for a germ whose domain is \mathcal{S} . Whenever \mathcal{C} is a subfamily of a category \mathcal{C} , one obtains a germ by considering the restriction of the product of \mathcal{C} to \mathcal{S} , that is, the partial binary operation \bullet on \mathcal{S} defined by $a \bullet b = ab$ whenever ab is defined in \mathcal{C} and it belongs to \mathcal{S} . In the other direction, starting with a germ, we can always construct a category.

Definition 3.10. If $\underline{\mathcal{S}}$ is a germ, we denote by $Cat(\underline{\mathcal{S}})$ the category $\langle \mathcal{S} | \mathcal{R}_{\bullet} \rangle^+$, where \mathcal{R}_{\bullet} is the family of all relations $a|b = a \bullet b$ with a, b in \mathcal{S} and $a \bullet b$ defined.

When we start with a category \mathcal{C} and a subfamily \mathcal{S} of \mathcal{C} , it may or may not be the case that the induced germ on \mathcal{S} contains enough information to reconstruct the initial category \mathcal{C} .

Example 3.11. Let us consider the braid monoid B_3^+ again. If we take $\mathcal{A} = \{1, \mathbf{a}, \mathbf{b}\}$, the table of the induced germ is shown aside, and the derived monoid is the free monoid based on \mathbf{a}, \mathbf{b} , hence one that is not isomorphic to B_3^+ .

\bullet	1	a	b
1	1	a	b
a		a	
b			b

By contrast, when we start with $\mathcal{S} = \{1, \mathbf{a}, \mathbf{b}, \mathbf{ab}, \mathbf{ba}, \Delta\}$, the table of the induced germ is less hollow, and the derived monoid is indeed isomorphic to B_3^+ .

\bullet	1	a	b	ab	ba	Δ
1	1	a	b	ab	ba	Δ
a	a		ab		Δ	
b	b	ba		Δ		
ab	ab	Δ				
ba	ba		Δ			
Δ	Δ					

The good point is that, as in the above example, the germ induced by a Garside family always contains enough information to reconstruct the initial category:

Lemma 3.12. [14, Proposition 4.8] *If \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} and $\underline{\mathcal{S}}$ is the germ induced on \mathcal{S} , the category $Cat(\underline{\mathcal{S}})$ is isomorphic to \mathcal{C} .*

3.3. Establishing left-cancellativity and recognizing Garside. Lemma 3.12 precisely shows that using a germ to specify a category is relevant here: if we start with a good candidate for a Garside family, the germ will indeed define the category.

Definition 3.13. A germ $\underline{\mathcal{S}}$ is said to be a *Garside germ* if \mathcal{S} embeds in $Cat(\underline{\mathcal{S}})$, the latter is left-cancellative, and (the image of) \mathcal{S} is a Garside family in that category.

So, for instance, the germs in Example 3.11 are Garside germs. By contrast, below is an example of a germ that is not a Garside germ although it defines the ambient monoid.

Example 3.14. Let $M = \langle a, b \mid ab = ba, a^2 = b^2 \rangle^+$, and \mathcal{S} consist of $1, a, b, ab, a^2$. The germ $\underline{\mathcal{S}}$ induced on \mathcal{S} is shown aside. It is left-associative and left-cancellative. The category (here the monoid) $Cat(\underline{\mathcal{S}})$ is (isomorphic to) M , as the relations $a|a = a^2 = b|b$ and $a|b = ab = b|a$ belong to the family \mathcal{R}_\bullet . However \mathcal{S} is not a Garside family in M , as a^3 admits no \mathcal{S} -normal decomposition: $a^2|a$ is not \mathcal{S} -greedy as ab left-divides a^3 but not a^2 , and $ab|b$ is not \mathcal{S} -greedy as a^2 left-divides a^3 but not ab .

\bullet	1	a	b	a^2	ab
1	1	a	b	a^2	ab
a	a	a^2	ab		
b	b	ab	a^2		
a^2	a^2				
ab	ab				

What we do below is to develop algorithms to decide whether a (finite) germ is a Garside germ. Note that this includes establishing that the defined category is left-cancellative. Our method is based on a result of [14].

Definition 3.15. Assume that $\underline{\mathcal{S}}$ is a germ.

(i) We define the *local left-divisibility* relation $\preccurlyeq_{\mathcal{S}}$ of $\underline{\mathcal{S}}$ by saying that $a \preccurlyeq_{\mathcal{S}} b$ holds if and only if there exists b' in \mathcal{S} satisfying $b = ab'$. We write $a \prec_{\mathcal{S}} b$ for the conjunction of $a \preccurlyeq_{\mathcal{S}} b$ and $b \not\preccurlyeq_{\mathcal{S}} a$, and we call a sequence a_1, \dots, a_n in \mathcal{S} *non-ascending* if $a_i \not\preccurlyeq_{\mathcal{S}} a_j$ holds for $1 \leq i < j \leq n$.

(ii) For $a_1|a_2$ in $\mathcal{S}^{[2]}$, we put $\mathcal{I}_{\mathcal{S}}(a_1, a_2) = \{b \in \mathcal{S} \mid a_1 \bullet b \text{ is defined and } b \preccurlyeq_{\mathcal{S}} a_2\}$.

Proposition 3.16. [14, Proposition 5.9] *A germ $\underline{\mathcal{S}}$ is a Garside germ if and only if it is left-associative, left-cancellative, and if, for any a_1, a_2 in \mathcal{S} there exists a $\preccurlyeq_{\mathcal{S}}$ -greatest element in $\mathcal{I}_{\mathcal{S}}(a_1, a_2)$ (that is, an element c in $\mathcal{I}_{\mathcal{S}}(a_1, a_2)$ such that $b \preccurlyeq_{\mathcal{S}} c$ holds for all b in $\mathcal{I}_{\mathcal{S}}(a_1, a_2)$).*

Corollary 3.17. *If \mathcal{S} is a Garside germ, if $a_1|a_2$ is in $\mathcal{S}^{[2]}$, if b is a $\preccurlyeq_{\mathcal{S}}$ -greatest element in $\mathcal{I}_{\mathcal{S}}(a_1, a_2)$, and if c is an element of \mathcal{S} satisfying $a_2 = b \bullet c$, then the path $a_1 b | c$ is an \mathcal{S} -normal decomposition of $a_1 a_2$ in $Cat(\underline{\mathcal{S}})$.*

Algorithm 3.18 (Recognizing a Garside germ).

Input: A finite germ $\underline{\mathcal{S}}$

Output: true if $\underline{\mathcal{S}}$ is a Garside germ, and false otherwise

```

1: isLeftCancellative,  $\preccurlyeq_{\mathcal{S}} := \text{LEFTDIVISIBILITY}(\mathcal{S})$ 
2: if not isLeftCancellative or not ISLEFTASSOCIATIVE( $\mathcal{S}$ ) then
3:   return false
4:  $\mathcal{S}' := \text{NONASCENDING}(\mathcal{S}, \preccurlyeq_{\mathcal{S}})$ 
5: for  $a_1|a_2$  in  $\mathcal{S}^{[2]}$  do
6:   if not JHASCREATESTELEMENT( $\mathcal{S}', a_1, a_2, \preccurlyeq_{\mathcal{S}}$ ) then
7:     return false
8: return true

9: function ISLEFTASSOCIATIVE( $\mathcal{S}$ )
10:  for  $a|b|c$  in  $\mathcal{S}^{[3]}$  do
11:    if  $(a \bullet b) \bullet c \downarrow$  and not  $b \bullet c \downarrow$  then
12:      return false
13:  return true

14: function LEFTDIVISIBILITY( $\mathcal{S}$ ) ▷ whether  $\mathcal{S}$  is left-cancellative and, if it is, a table
    (denoted by  $\preccurlyeq_{\mathcal{S}}$ ) with the truth values  $\llbracket a \preccurlyeq_{\mathcal{S}} b \rrbracket$  of  $a \preccurlyeq_{\mathcal{S}} b$  for  $a, b$  in  $\mathcal{S}$ 
15:  for  $a \in \mathcal{S}$  do
16:    for  $b \in \mathcal{S}$  do
17:       $\llbracket a \preccurlyeq_{\mathcal{S}} b \rrbracket := \text{false}$ 
18:    for  $b \in \mathcal{S}$  do
19:      if  $a \bullet b \downarrow$  then
20:        if  $\llbracket a \preccurlyeq_{\mathcal{S}} a \bullet b \rrbracket$  then
21:          return false
22:        else
23:           $\llbracket a \preccurlyeq_{\mathcal{S}} a \bullet b \rrbracket := \text{true}$ 

```

```

24:   return true,  $\preceq_{\mathcal{S}}$ 
25: function NONASCENDING( $\mathcal{S}$ ,  $\preceq_{\mathcal{S}}$ ) ▷  $\mathcal{S}$  as a non-ascending sequence
26:    $\mathcal{S}' := []$ 
27:   for  $a \in \mathcal{S}$  do
28:     if  $\exists i \in \{1, \dots, |\mathcal{S}'|\}$  with  $\mathcal{S}'[i] \prec_{\mathcal{S}} a$  then
29:       insert  $a$  into  $\mathcal{S}$  at position  $\min(\{i \in \{1, \dots, |\mathcal{S}'|\} \mid \mathcal{S}'[i] \prec_{\mathcal{S}} a\})$ 
30:     else
31:       append  $a$  to  $\mathcal{S}'$ 
32:   return  $\mathcal{S}'$ 
33: function JHASGREATESTELEMENT( $\mathcal{S}'$ ,  $a_1$ ,  $a_2$ ,  $\preceq_{\mathcal{S}}$ ) ▷  $\mathcal{S}'$  non-ascending
34:    $c := \perp$ 
35:   for  $i := 1$  to  $|\mathcal{S}'|$  do
36:     if  $a_1 \bullet \mathcal{S}'[i] \downarrow$  and  $\mathcal{S}'[i] \preceq_{\mathcal{S}} a_2$  then ▷  $\mathcal{S}'[i] \in \mathcal{J}_{\mathcal{S}}(a_1, a_2)$ 
37:       if  $c = \perp$  then
38:          $c := \mathcal{S}'[i]$ 
39:       else if not  $\mathcal{S}'[i] \preceq_{\mathcal{S}} c$  then
40:         return false
41:   return true

```

Proposition 3.19. *Assume that $\underline{\mathcal{S}}$ is a finite germ with $|\mathcal{S}| = n$ and that the partial binary operation \bullet can be computed in time $O(1)$. Then the following hold; cf. Algorithm 3.18.*

- (i) *The function `IsLeftAssociative` decides in time $O(n^3)$ whether $\underline{\mathcal{S}}$ is left-associative.*
- (ii) *The function `LeftDivisibility` decides whether $\underline{\mathcal{S}}$ is left-cancellative and, if it is, computes the left-divisibility relation $\preceq_{\mathcal{S}}$ on \mathcal{S} with respect to \bullet in time $O(n^2)$.*
- (iii) *Given the left-divisibility relation $\preceq_{\mathcal{S}}$, the function `NonAscending` computes a non-ascending sequence containing the elements of \mathcal{S} in time $O(n^2)$.*
- (iv) *Given $a_1|a_2$ in $\mathcal{S}^{[2]}$, a non-ascending sequence \mathcal{S}' containing the elements of \mathcal{S} , and the left-divisibility relation $\preceq_{\mathcal{S}}$, the function `JHasGreatestElement` decides in time $O(n)$ whether $\mathcal{J}_{\mathcal{S}}(a_1, a_2)$ has a $\preceq_{\mathcal{S}}$ -greatest element.*
- (v) *It is decidable in time $O(n^3)$ whether \mathcal{S} is a Garside germ.*

Proof. Claims (i) and (ii) are obvious from the pseudocode in Algorithm 3.18 and the definitions.

The sequence \mathcal{S}' constructed in the function `NonAscending` is non-ascending at any time by induction: If there is no i satisfying $\mathcal{S}'[k] \prec_{\mathcal{S}} a$, the induction step is trivial. Otherwise, one has $\mathcal{S}'[j] \not\prec_{\mathcal{S}} a$ for all $j < i$ by the choice of i , and $a \not\prec_{\mathcal{S}} \mathcal{S}'[j]$ for $j \geq i$, as $\mathcal{S}'[k] \prec_{\mathcal{S}} a \prec_{\mathcal{S}} \mathcal{S}'[j]$ would contradict the induction hypothesis. It is clear from the pseudocode that the time complexity is $O(n^2)$, so claim (iii) holds.

The function `JHasGreatestElement` tests the elements of \mathcal{S} for membership in $\mathcal{J}_{\mathcal{S}}(a_1, a_2)$ in non-ascending order. Moreover, the set $\mathcal{J}_{\mathcal{S}}(a_1, a_2)$ is non-empty, as it contains ε_x , where x is the target of a_1 . Hence, the set $\mathcal{J}_{\mathcal{S}}(a_1, a_2)$ has a $\preceq_{\mathcal{S}}$ -greatest element if and only if the *first* encountered element is an upper bound; this is what the function `JHasGreatestElement` tests. It is clear from the pseudocode that the time complexity is $O(n)$, so claim (iv) holds.

Claim (v) then follows with Proposition 3.16. □

Example 3.20. We apply Algorithm 3.18 to the second germ of Example 3.11, that is, we have $\mathcal{S} = \{1, \mathbf{a}, \mathbf{b}, \mathbf{ab}, \mathbf{ba}, \Delta\}$ and \bullet is the partial binary operation on \mathcal{S} induced by the multiplication in B_3^+ .

One readily verifies that the germ is left-associative and left-cancellative, and computes the left-divisibility relation $\preceq_{\mathcal{S}}$ which is given in the left table below.

We obtain $\mathcal{S}' = (\Delta, \mathbf{ab}, \mathbf{ba}, \mathbf{b}, \mathbf{a}, 1)$ and non-ascending sequences describing the sets $\mathcal{J}_{\mathcal{S}}(a_1, a_2)$ given in the right table. For each set $\mathcal{J}_{\mathcal{S}}(a_1, a_2)$, the first listed element is a maximum, showing that \mathcal{S} is a Garside germ.

\mathcal{J}_S	1	a	b	ab	ba	Δ
1	\mathcal{J}	\mathcal{J}	\mathcal{J}	\mathcal{J}	\mathcal{J}	\mathcal{J}
a		\mathcal{J}		\mathcal{J}		\mathcal{J}
b			\mathcal{J}		\mathcal{J}	\mathcal{J}
ab				\mathcal{J}		\mathcal{J}
ba					\mathcal{J}	\mathcal{J}
Δ						\mathcal{J}

\mathcal{J}_S	1	a	b	ab	ba	Δ
1	(1)	(a, 1)	(b, 1)	(ab, a, 1)	(ba, b, 1)	\mathcal{S}'
a	(1)	(1)	(b, 1)	(1)	(ba, b, 1)	(ba, b, 1)
b	(1)	(a, 1)	(1)	(ab, a, 1)	(1)	(ab, a, 1)
ab	(1)	(a, 1)	(1)	(a, 1)	(1)	(a, 1)
ba	(1)	(1)	(b, 1)	(1)	(b, 1)	(b, 1)
Δ	(1)	(1)	(1)	(1)	(1)	(1)

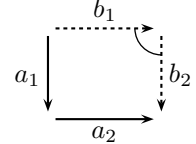
3.4. Further questions. From that point, results are similar to those of Subsection 2.4: once a germ has been shown to be a Garside germ, it is known to be a Garside family in its ambient category, and, for instance, Propositions 2.28 and 2.29 directly apply for the existence of common multiples. The only point worth mentioning here is right-cancellativity: let us observe that a germ is a symmetric structure, so applying the method of Subsection 3.3 to the opposite of the considered germ $\underline{\mathcal{S}}$, that is, $(\mathcal{S}, \tilde{\bullet})$ with $\tilde{\bullet}$ defined by $a \tilde{\bullet} b = b \bullet a$, directly leads to a right-cancellativity criterion.

4. COMPUTING WITH A GARSIDE FAMILY, POSITIVE CASE

We now turn to a different series of questions: here we no longer aim at deciding whether the ambient category and a candidate family are relevant for the Garside approach, but we assume that they are and we aim at computing in the context so obtained. In this section, we consider computations that take place inside the considered category, and postpone to the next section the computations that take place in the groupoid of fractions. We shall successively consider computing normal decompositions (Subsections 4.1 and 4.2), solving the Word Problem (Subsection 4.3) and computing least common multiples (Subsection 4.4).

4.1. Computing normal decompositions from the left. By definition, a Garside family gives rise to distinguished decompositions for the elements of the ambient category, namely \mathcal{S} -normal decompositions. The first natural algorithmic question is to determine an \mathcal{S} -normal decomposition of an element g specified by an arbitrary decomposition in terms of \mathcal{S}^\sharp . So the question is, starting from an \mathcal{S}^\sharp -path w , to find an equivalent \mathcal{S} -normal path. Here we shall use an incremental approach, starting from a solution in the case of length two paths.

Definition 4.1. A subfamily \mathcal{A} of a left-cancellative category \mathcal{C} satisfies *Property* \square if, for every $a_1|a_2$ in $\mathcal{A}^{[2]}$, there exists an \mathcal{A} -greedy decomposition $b_1|b_2$ of a_1a_2 with b_1 and b_2 in \mathcal{A} . In this case, a map W that chooses, for every $a_1|a_2$ in $\mathcal{A}^{[2]}$, a pair (b_1, b_2) as above is called a \square -witness on \mathcal{A} .



If \mathcal{S} is a Garside family, then one easily shows that there exists a \square -witness on \mathcal{S}^\sharp , that is, every element of $(\mathcal{S}^\sharp)^2$ admits an \mathcal{S} -normal decomposition of length 2. Moreover, if the ambient category admits no nontrivial invertible element, then, by Proposition 1.11, the \square -witness on \mathcal{S}^\sharp is unique. Note that, if the Garside family was initially specified as a germ, then Corollary 3.17 directly provides a \square -witness; see Table 1 for an example.

Algorithm 4.2 (Left-multiplication—see Figure 6).

Context: A Garside family \mathcal{S} in a left-cancellative category \mathcal{C} , a \square -witness W for \mathcal{S}^\sharp

Input: An element b of \mathcal{S}^\sharp and an \mathcal{S} -normal decomposition $a_1|\dots|a_p$ of an element g of \mathcal{C} such that bg exists

Output: An \mathcal{S} -normal decomposition of bg

- 1: $b_0 := b$
- 2: **for** i increasing from 1 to p **do**
- 3: $(a'_i, b_i) := W(b_{i-1}, a_i)$
- 4: **return** $a'_1|\dots|a'_p|b_p$

	ε	a	b	ab	ba	Δ
ε	(1, 1)	(a, 1)	(b, 1)	(ab, 1)	(ba, 1)	(Δ , 1)
a	(a, 1)	(a, a)	(ab, 1)	(a, ab)	(Δ , 1)	(Δ , b)
b	(b, 1)	(ba, 1)	(b, b)	(Δ , 1)	(b, ba)	(Δ , a)
ab	(ab, 1)	(Δ)	(ab, b)	(Δ , b)	(ab, ba)	(Δ , ba)
ba	(ba, 1)	(ba, a)	(Δ , 1)	(ba, ab)	(Δ , a)	(Δ , ab)
Δ	(Δ , 1)	(Δ , a)	(Δ , b)	(Δ , ab)	(Δ , ba)	(Δ , Δ)

TABLE 1. The \square -witness on the Garside family $\{1, a, b, ab, ba, \Delta\}$ in the braid monoid B_3^+ : for every pair of elements (a_1, a_2) , it specifies the unique normal decomposition of $a_1 a_2$ of length 2. The values directly follow from considering \mathcal{S} as a germ and using the maximal \mathcal{J} -function of Example 3.20.

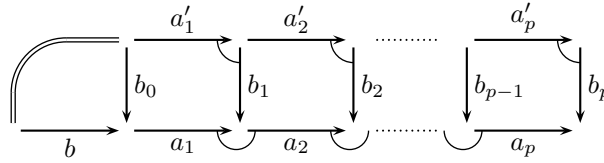


FIGURE 6. Algorithm 4.2: starting from b in \mathcal{S}^\sharp and an \mathcal{S} -normal decomposition of g , it returns an \mathcal{S} -normal decomposition of bg .

Proposition 4.3. *Assume that \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} , and W is a \square -witness for \mathcal{S}^\sharp . Then Algorithm 4.2 running on b and an \mathcal{S} -normal decomposition $a_1 | \dots | a_p$ of g returns an \mathcal{S} -normal decomposition of bg . The function W is called p times.*

Proof. By construction, the diagram of Figure 6 is commutative. By assumption, $a_i | a_{i+1}$ is \mathcal{S} -greedy for every i , and, by the defining property of a \square -witness, $a'_i | b_i$ is \mathcal{S} -normal for every i . Then the first domino rule (Lemma 1.16) implies that $a'_i | a'_{i+1}$ is \mathcal{S} -normal. It is obvious from Algorithm 4.2 that the function W is invoked p times. \square

We can now compute \mathcal{S} -normal decompositions for an arbitrary element of \mathcal{C} specified by an arbitrary \mathcal{S}^\sharp -decomposition by iterating Algorithm 4.2.

Algorithm 4.4 (Normal decomposition).

Context: A Garside family \mathcal{S} in a left-cancellative category \mathcal{C} , a \square -witness W for \mathcal{S}^\sharp

Input: An \mathcal{S}^\sharp -decomposition $b_1 | \dots | b_p$ of an element g of \mathcal{C}

Output: An \mathcal{S} -normal decomposition of g

- 1: $b_{0,i} := b_i$ for $1 \leq i \leq p$
- 2: **for** j decreasing from p to 1 **do**
- 3: **for** i increasing from 1 to $p - j$ (if any) **do**
- 4: $(a_{i,j-1}, b_{i,j}) := W(b_{i-1,j}, a_{i,j})$
- 5: $a_{p-j+1,j-1} := b_{p-j,j}$
- 6: **return** $a_{1,0} | \dots | a_{p,0}$

Proposition 4.5. *Assume that \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} and W is a \square -witness for \mathcal{S}^\sharp . Then Algorithm 4.4 running on an \mathcal{S}^\sharp -decomposition $b_1 | \dots | b_p$ of an element g of \mathcal{C} returns an \mathcal{S} -normal decomposition of g . The map W is appealed to $p(p-1)/2$ times.*

Proof. Consider the **for**-loop for j and assume that before the execution of its body, $a_{1,j} | \dots | a_{p-j,j}$ is an \mathcal{S} -normal decomposition of $b_{0,j+1} \dots b_{0,p}$, which is also $b_{j+1} \dots b_p$; this condition is trivially satisfied for $j = p$.

According to Proposition 4.3, the execution of lines 3-5 involves $p-j$ invocations of the map W and produces an \mathcal{S} -normal decomposition $a_{1,j-1}|\cdots|a_{p-j+1,j-1}$ of $b_{0,j}\cdots b_{0,p}$ that is, of $b_j\cdots b_p$. The claim then follows by induction. \square

Example 4.6. Consider the Garside family $\mathcal{S} = \{1, \mathbf{a}, \mathbf{b}, \mathbf{ab}, \mathbf{ba}, \Delta\}$ and the element $g = \mathbf{bab}^2$. We begin with the empty word, \mathcal{S} -normal decomposition of 1, and, starting from the right, we multiply by the successive letters on the left using Algorithm 4.2 and the \square -witness of Table 1, thus finding \mathbf{b} , $\mathbf{b|b}$, $\mathbf{ab|b}$ and, finally, $\Delta|\mathbf{b}$ for the (unique) \mathcal{S} -normal decompositions of \mathbf{b} , \mathbf{b}^2 , \mathbf{ab}^2 , and g .

4.2. Computing normal decompositions using right-multiplication. Algorithm 4.2 is not symmetric: the construction starts from the left and moves to the right. It is natural to wonder whether a symmetric right-to-left version exists. The answer depends on the considered Garside family \mathcal{S} : if the latter satisfies the second domino rule (Lemma 1.31), the previous constructions have counterparts based on right-multiplication.

Algorithm 4.7 (Right-multiplication—see Figure 7).

Context: A Garside family \mathcal{S} satisfying the second domino rule in a left-cancellative category \mathcal{C} , a \square -witness W for \mathcal{S}^\sharp

Input: An element b of \mathcal{S}^\sharp , an \mathcal{S} -normal decomposition $a_1|\cdots|a_p$ of an element g of \mathcal{C} such that gb is defined

Output: An \mathcal{S} -normal decomposition of gb

- 1: $b_p := b$
- 2: **for** i decreasing from p to 1 **do**
- 3: $(b_{i-1}, a'_i) := W(a_i, b_i)$
- 4: **return** $b_0|a'_1|\cdots|a'_p$

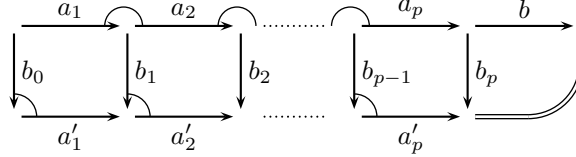


FIGURE 7. Algorithm 4.7: starting from an \mathcal{S} -normal decomposition of g and b , it returns an \mathcal{S} -normal decomposition of gb .

Proposition 4.8 (right-multiplication). *Assume that \mathcal{S} is a Garside family satisfying the second domino rule in a left-cancellative category \mathcal{C} and W is a \square -witness for \mathcal{S}^\sharp . Then Algorithm 4.7 running on b in \mathcal{S}^\sharp and an \mathcal{S} -normal decomposition $a_1|\cdots|a_p$ of g in \mathcal{C} returns an \mathcal{S} -normal decomposition of gb , if the latter is defined. The function W is invoked p times.*

Proof. The commutativity of the diagram gives $b_0a'_1\cdots a'_p = a_1\cdots a_pb$. Applying the second domino rule to each two-square subdiagram of the diagram of Figure 7 starting from the right, we see that the sequence $b_0|a'_1|\cdots|a'_p$ is \mathcal{S} -greedy. As all entries lie in \mathcal{S}^\sharp , the sequence is \mathcal{S} -normal. \square

By iterating Algorithm 4.7 (when applicable), we easily obtain a symmetric version of Algorithm 4.4 and Proposition 4.5, that we shall not explicitly state.

Remark 4.9. The effect of the second domino rule is to shorten the computation of certain normal decompositions. Indeed, assume that $b_1|\cdots|b_q$ and $a_1|\cdots|a_p$ are \mathcal{S} -normal paths and b_qa_1 is defined. By applying Proposition 4.8, we can compute an \mathcal{S} -normal decomposition of the product $b_1\cdots b_qa_1\cdots a_p$ by filling a diagram as in Figure 8. When valid, the second domino rule guarantees that the path consisting of the first p top edges followed by q vertical edges is \mathcal{S} -normal, that is, the triangular part of the diagram may be forgotten.

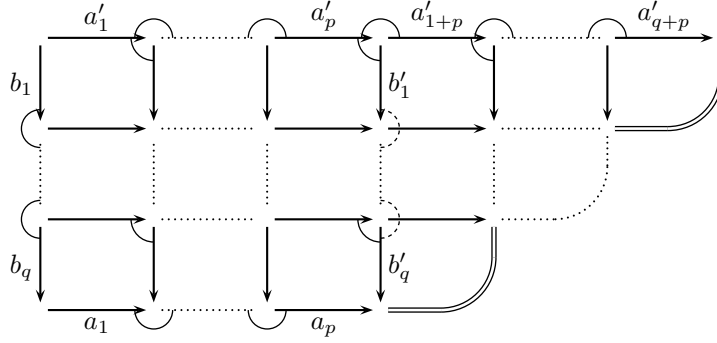
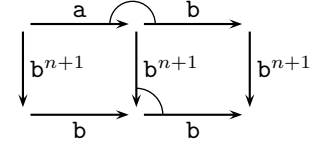


FIGURE 8. Finding an \mathcal{S} -normal decomposition of $b_1 \cdots b_q a_1 \cdots a_p$ when $b_1 | \cdots | b_q$ and $a_1 | \cdots | a_p$ are \mathcal{S} -normal: using Proposition 4.3, hence the first domino rule only, one determines the \mathcal{S} -normal sequence $a'_1 | \cdots | a'_{q+p}$ in $pq + q(q-1)/2$ steps; if the second domino rule is valid, the sequence $b'_1 | \cdots | b'_q$ is already \mathcal{S} -normal, and $a'_1 | \cdots | a'_p | b'_1 | \cdots | b'_q$ is an \mathcal{S} -normal decomposition of $b_1 \cdots b_q a_1 \cdots a_p$.

We conclude with an example of a Garside family that does not satisfy the second domino rule: its existence shows that no uniform result for multiplication on the right is possible.

Example 4.10. For $n \geq 2$, let $M_n = \langle \mathbf{a}, \mathbf{b} \mid \mathbf{a}\mathbf{b}^n = \mathbf{b}^{n+1} \rangle^+$. The method of Subsection 2.2 shows that M_n is left-cancellative, and that any two elements of M_n that admit a common right-multiple admit a right-lcm. On the other hand, M_n is not right-cancellative since we have $\mathbf{a}\mathbf{b}^{n-1} \neq \mathbf{b}^n$ and $\mathbf{a}\mathbf{b}^n = \mathbf{b}^{n+1}$. Let $S_n = \{1, \mathbf{a}, \mathbf{b}, \mathbf{b}^2, \dots, \mathbf{b}^{n+1}\}$, a subset of M_n with $n+3$ elements. Using Lemma 3.1, one can check that S_n is a Garside family in M_n .

Now the second domino rule is not valid for S_n in M_n . Indeed, the diagram aside is commutative, the paths $\mathbf{a}|\mathbf{b}$ and $\mathbf{b}^{n+1}|\mathbf{b}$ are S_n -greedy, and all edges corresponds to elements of S_n . However $\mathbf{b}|\mathbf{b}$ is not S_n -greedy since \mathbf{b}^2 lies in S_n .



4.3. Solving the Word Problem. We shall describe two solutions of the Word Problem for a left-cancellative category equipped with a Garside family, one based on normal decompositions, and one based on reversing.

Using \mathcal{S} -normal decompositions. Whenever an effective method determining an \mathcal{S} -normal decomposition is available, a solution to the Word Problem is very close. However, due to the possible existence of nontrivial invertible elements, \mathcal{S} -normal decompositions need not be readily unique, so a final comparison step is needed.

Definition 4.11. Assume that \mathcal{C} is a left-cancellative category and \mathcal{A} is included in \mathcal{C} . An $=^\times$ -test on \mathcal{A} is a map E of \mathcal{A}^2 to $\mathcal{C}^\times \cup \{\perp\}$ satisfying $\mathbf{a}E(\mathbf{a}, \mathbf{b}) = \mathbf{b}$ whenever \mathbf{a}, \mathbf{b} are $=^\times$ -equivalent, and $E(\mathbf{a}, \mathbf{b}) = \perp$ otherwise.

Algorithm 4.12 (Word Problem, positive case I).

Context: A left-cancellative category \mathcal{C} , a Garside subfamily \mathcal{S} of \mathcal{C} , a \square -witness W on \mathcal{S}^\sharp , an $=^\times$ -test E on \mathcal{S}^\sharp

Input: Two \mathcal{S}^\sharp -paths u, v

Output: `true` if u, v represent the same element of \mathcal{C} , and `false` otherwise

- 1: **if** `src(u) \neq src(v)` or `trg(u) \neq trg(v)` **then**
- 2: **return false**
- 3: **else**
- 4: use Algorithm 4.4 with W to find an \mathcal{S} -normal path $a_1 | \cdots | a_p$ representing $[u]$
- 5: use Algorithm 4.4 with W to find an \mathcal{S} -normal path $b_1 | \cdots | b_q$ representing $[v]$
- 6: **return** the value of `COMPARENORMALPATHS($a_1 | \cdots | a_p, b_1 | \cdots | b_q$)`

```

7: function COMPARENORMALPATHS( $a_1 | \dots | a_p, b_1 | \dots | b_q$ )
    $\triangleright$  the paths should have the same source and the same target
8:    $x :=$  source of  $b_1$ ;  $y :=$  target of  $b_q$ 
9:    $e_0 := 1_x$ ;  $b_i := 1_y$  for  $q < i \leq \max(q, p)$ ;  $a_j := 1_y$  for  $p < j \leq \max(q, p)$ 
10:  for  $k := 1$  to  $\max(q, p)$  do
11:    if  $E(b_i, e_{i-1}a_i) \neq \perp$  then
12:       $e_i := E(b_i, e_{i-1}a_i)$ 
13:    else
14:      return false
15:  return  $\llbracket e_i = 1_y \rrbracket$ 

```

Proposition 4.13. *Assume that \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} , W is a \square -witness on \mathcal{S}^\sharp , and E is an $=^x$ -test on \mathcal{S}^\sharp .*

(i) *Given two \mathcal{S} -normal paths u and v of length at most ℓ , the function COMPARENORMALPATHS in Algorithm 4.12 decides in time $O(\ell)$ whether $[u] = [v]$ holds.*

(ii) *Given two \mathcal{S}^\sharp -paths u and v of length at most ℓ , Algorithm 4.12 decides in time $O(\ell^2)$ whether $[u] = [v]$ holds.*

Proof. (i) By Proposition 1.11, two \mathcal{S} -normal paths u and v satisfy $[u] = [v]$ if and only if they are \mathcal{C}^x -deformations of one another. The latter in turn is the case if and only if $E(b_i, e_{i-1}a_i) \neq \perp$ for all $i = 1, \dots, \max(p, q)$ and $e_{\max(p, q)} = 1_y$ hold in lines 10-14 of the function COMPARENORMALPATHS. It is obvious that this test takes time $O(\ell)$.

(ii) If u and v have different sources or different targets, clearly we have $[u] \neq [v]$. Otherwise, one has $[u] = [v]$ if and only if the paths computed in lines 4 and 5 of Algorithm 4.12 represent the same element. By Proposition 4.5, lines 4 and 5 of Algorithm 4.12 have a cost of $O(\ell^2)$ and the lengths of the produced paths is at most ℓ . The claim then follows with (i). \square

Example 4.14. Consider once again the 3-strand braid monoid B_3^+ . Let \mathcal{S} be $\{1, \mathbf{a}, \mathbf{b}, \mathbf{ab}, \mathbf{ba}, \Delta\}$. We saw in Example 2.25 that \mathcal{S} is a (minimal) Garside family in B_3^+ containing 1, and determined in Table 1 a \square -witness on \mathcal{S} . As B_3^+ contains no nontrivial invertible element, testing for $=^x$ reduces to testing equality.

As in Figure 5, consider $u = \mathbf{a|b|b}$ and $v = \mathbf{b|a|b|b}$. Applying Algorithm 4.4, we obtain the normal decomposition $\mathbf{ab|b}$ of $[u]$ and the normal decomposition $\Delta|\mathbf{b}$ of $[v]$. As \mathbf{ab} and Δ are not equal, hence not $=^x$ -equivalent, Algorithm 4.12 returns **false**, that is, one has $[u] \neq [v]$.

Using reversing. An alternative solution of the Word Problem that does not require computing a normal form can be given using the reversing method of Subsection 2.1. By definition, if the investigated category \mathcal{C} is specified using a right-complemented presentation $(\mathcal{A}; \mathcal{R})$ for which right-reversing is complete, then two \mathcal{A} -paths u, v represent the same element of \mathcal{C} if and only if the signed path $\bar{u}|v$ reverses to an empty path, that is, if and only if Algorithm 2.5 running on $\bar{u}|v$ returns an empty path. This however provides a solution to the Word Problem only if right-reversing is known to terminate in finite time. By Lemma 2.9, this happens whenever any two elements with the same source admit a common right-multiple in the considered category, but nothing can be said in more general cases.

Now the termination problem does not arise when one considers a presentation that is associated with a short right-complement, in which case the Word Problem can be solved using reversing. The point here is that every Garside family gives rise to such a presentation:

Lemma 4.15. *Under the assumptions of Proposition 1.15, the obtained presentation $(\mathcal{S}; \mathcal{R})$ of \mathcal{C} is associated with a short right-complement, and right-reversing is complete for this presentation.*

Proof. Let θ be the partial map defined on \mathcal{S} by $\theta(a, b) = a \setminus b$. By definition, θ is a right-complement on \mathcal{S} and, by Lemma 1.14, it is short. Next, the associativity of the right-lcm operation implies that θ satisfies the cube condition for every triple of elements of \mathcal{S} . Then Proposition 2.18, implies that right-reversing is complete for $(\mathcal{S}; \mathcal{R})$. \square

It follows that the presentation $(\mathcal{S}; \mathcal{R})$ is eligible for right-reversing and that the latter solves the Word Problem.

Algorithm 4.16 (Word Problem, positive case II).

Context: A Garside family \mathcal{S} in a left-cancellative category \mathcal{C} that admits unique local right-lcms

Input: Two \mathcal{S} -paths u, v

Output: **true** if u, v represent the same element of \mathcal{C} , and **false** otherwise

1: $\theta :=$ the right-lcm selector on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$

2: **ret** := the return value of Algorithm 2.3 for right-complement θ and input $\overline{u|v}$

3: **return** $\llbracket \text{ret} = \varepsilon_- \rrbracket$ \triangleright **ret** is either a positive-negative path or equal to **fail**

Lemmas 4.15 and 2.4 immediately imply:

Proposition 4.17. *Assume that \mathcal{S} is a Garside family in a left-cancellative category \mathcal{C} that admits unique local right-lcms.*

(i) *Algorithm 4.16 solves the Word Problem of \mathcal{C} with respect to \mathcal{S} .*

(ii) *If \mathcal{S} is finite, the complexity of Algorithm 4.16 is quadratic in the length of the input paths.*

Example 4.18. Consider once again the 3-strand braid monoid B_3^+ . As seen in Example 2.25, the five elements family $\{a, b, ab, ba, \Delta\}$ is a (minimal) Garside family in B_3^+ . The resulting presentation of B_3^+ is $(a, b, ab, ba, \Delta; \mathcal{R})$, where \mathcal{R} consists of $\binom{5}{2}$ relations (in which we leave the concatenation sign to avoid ambiguities) $a|ba = b|ab$, $a|b = ab$, $a|ba = ba|b$, $a|ba = \Delta$, $b|ab = ab|a$, $b|a = ba$, $b|ab = \Delta$, $ab|a = ba|b$, $ab|a = \Delta$, $ba|b = \Delta$. Starting, as in Example 4.14, from the words $u = a|b|b$ and $v = b|a|b|b$, we apply Algorithm 2.3 for the input $\overline{u|v}$, obtaining $w = ab|\overline{a}$ (see Figure 9) and thus, as w is not empty, we conclude once again that u and v are not equivalent.

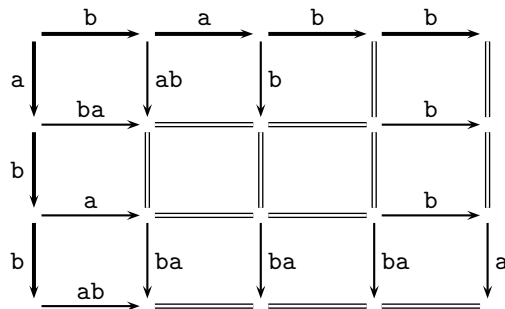


FIGURE 9. The grid associated with the right-reversing of $\overline{u|v}$ in Example 4.18; compare with the grid of Figure 5: the final paths are equivalent, but not equal, which is not contradictory as they are obtained in different ways.

One of the benefits of the reversing approach is to provide at the same time a decision method for the left-divisibility relation, as Lemma 4.15 also implies:

Proposition 4.19. *If line 3 of Algorithm 4.16 is modified so that the algorithm returns **true** if **ret** is a positive path, and **false** otherwise, the modified algorithm returns **true** if and only if $[u]$ left-divides $[v]$ in \mathcal{C} .*

Remark 4.20. The above results extend to every left-cancellative category, possibly containing nontrivial invertible elements and not admitting local right-lcms, at the expense of considering a more general, non-deterministic version of reversing. But, as said in Remark 2.7, we do not consider such extensions here.

4.4. Computing lcms.

Right-lcms. It follows from Lemma 2.22 that, whenever a right-complemented presentation $(\mathcal{A}; \mathcal{R})$ is given for a left-cancellative category \mathcal{C} and right-reversing is complete for that presentation, then right-reversing provides a direct algorithm for computing right-lcms when they exist.

Example 4.21. Consider again the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{aba} = \mathbf{bab})$ of the braid monoid B_3^+ and the elements \mathbf{ab}^2 and \mathbf{bab}^2 . We saw in Example 2.20 that right-reversing is complete for this presentation. Now, we saw in Figure 5 that the word $\overline{\mathbf{b}}|\overline{\mathbf{b}}|\overline{\mathbf{a}}|\mathbf{b}|\mathbf{a}|\mathbf{b}|\mathbf{b}$ is right-reversible to $\mathbf{a}|\mathbf{b}|\overline{\mathbf{a}}$ and, therefore, we conclude that the right-lcm of \mathbf{ab}^2 and \mathbf{bab}^2 is $\mathbf{ab}^2 \cdot \mathbf{ab}$ (which is also $\mathbf{bab}^2 \cdot \mathbf{a}$).

Now, independently of the way the considered category \mathcal{C} was initially specified, if one happens to know a right-lcm selector θ on a generating subfamily \mathcal{A} , we can use the latter, which by definition is a right-complement, to right-reverse \mathcal{A} -paths. If the right-selector is not short, termination is not guaranteed but, if it is short, we immediately obtain:

Proposition 4.22. *Assume that θ is a short right-lcm selector on some generating family \mathcal{A} in a left-cancellative category \mathcal{C} that admits local right-lcms. For all \mathcal{A} -paths u, v , if Algorithm 2.3 running on $\overline{u}|v$ returns a positive–negative path $v'|\overline{u}'$, then $[u|v']$ is a right-lcm of $[u]$ and $[v]$ in \mathcal{C} ; if it returns **fail**, then $[u]$ and $[v]$ admit no right-lcm in \mathcal{C} . The complexity of the computation is in $O((\|u\| + \|v\|)^2)$*

Proof. As observed in the proof of Lemma 4.15, right- θ -reversing must be complete in this case, and Lemma 2.22 then gives the expected result. Alternatively, we can directly observe that, if $g_1 f'$ is a right-lcm of f and g_1 and $g_2 f''$ is a right-lcm of f' and g_2 , then $g_1 g_2 f''$ is a right-lcm of f and $g_1 g_2$, which inductively implies that the diagonal of every rectangle in a right-reversing grid represents the right-lcm of the left and top edges. \square

The point here is that, by Lemma 4.15, every Garside family in a left-cancellative category that admits right-lcms gives rise to a short right-lcm selector, and therefore is eligible for Proposition 4.22. Moreover, by Proposition 2.30, the existence of right-lcms in \mathcal{C} reduces to the existence of right-lcms inside \mathcal{S} . It follows that, when \mathcal{S}^\sharp is finite, one can effectively decide the existence of right-lcms for the elements of \mathcal{S}^\sharp and, in addition, obtain a right-lcm selector.

Example 4.23. Starting with the Garside family $\{\mathbf{a}, \mathbf{b}, \mathbf{ab}, \mathbf{ba}, \Delta\}$ in B_3^+ , the right-lcm selector θ on \mathcal{S} leads to the presentation of Example 4.18. Considering as in Example 4.21 the elements \mathbf{ab}^2 and \mathbf{bab}^2 , we find their right-lcm by right- θ -reversing $\overline{\mathbf{b}}|\overline{\mathbf{b}}|\overline{\mathbf{a}}|\mathbf{b}|\mathbf{a}|\mathbf{b}|\mathbf{b}$. As seen in Figure 9, the latter word reverses to $\mathbf{a}|\mathbf{b}|\overline{\mathbf{a}}$, and we conclude once again that the right-lcm of \mathbf{ab}^2 and \mathbf{bab}^2 is $\mathbf{ab}^2 \cdot \mathbf{ab}$.

Remark 4.24. The above approach consisting in using the right-reversing transformation associated with a (short or non-short) right-lcm selector θ on some subfamily \mathcal{A} may work even when \mathcal{A} is not a Garside family, provided \mathcal{A} generates the ambient category \mathcal{C} and the family of all relations $a\theta(a, b) = b\theta(b, a)$ with a, b in \mathcal{A} makes a presentation of \mathcal{C} . Interestingly, this is always the case when \mathcal{C} is right-Noetherian, but the example of the Dubrovina-Dubrovin 4-strand braid monoid [13] shows that the property may fail in the general case.

Left-lcms. The case of left-lcms is symmetric, and we shall not say much: by the counterpart of Proposition 4.22, if there exists a short left-lcm-selector $\tilde{\theta}$ on a generating family \mathcal{A} , then left- $\tilde{\theta}$ -reversing a positive–negative path $v|\overline{u}$ possibly leads to a negative–positive path $\overline{u}'|v'$ such that $u'|v$ and $v'|u$ represent the left-lcm of $[u]$ and $[v]$.

5. COMPUTING WITH A STRONG GARSIDE FAMILY, SIGNED CASE

In this final section, we show how to use (strong) Garside families to compute in groupoids of fractions. The successive problems we address are finding symmetric normal decompositions (Subsection 5.1), finding Δ -normal decompositions (Subsection 5.2), solving the Word Problem (Subsection 5.3), computing inverses (Subsection 5.4) and, finally, computing lower and upper bounds (Subsection 5.5).

5.1. Computing symmetric normal decompositions. According to Proposition 1.21, if \mathcal{S} is a Garside family in a left-Ore category \mathcal{C} , then every element of $\mathcal{E}nv(\mathcal{C})$, the groupoid of fractions of \mathcal{C} , that can be represented as a right-fraction admits a symmetric \mathcal{S} -normal decomposition. Here we address the question of algorithmically computing such a decomposition.

Starting from a right fraction. Owing to the above recalled restriction on the considered elements (namely that of being expressible as a right-fraction), it is natural to first start with a positive–negative path. Then, Lemma 1.20 reduces the computation of a symmetric normal decomposition to the determination of a left-lcm:

Algorithm 5.1 (Symmetric normal decomposition, positive–negative input).

Context: A Garside family \mathcal{S} in a left-Ore category \mathcal{C} that admits left-lcms, a \square -witness W on \mathcal{S}^\sharp , a procedure computing left-lcms in \mathcal{C}

Input: A positive–negative \mathcal{S}^\sharp -path $v|\bar{u}$

Output: A symmetric \mathcal{S} -normal decomposition of the element $[v][u]^{-1}$ of $\mathcal{E}nv(\mathcal{C})$

- 1: find u', v' such that $u'|v$ and $v'|u$ represent a left-lcm of $[u]$ and $[v]$
- 2: find an \mathcal{S} -normal path $b_1|\dots|b_q$ equivalent to u' using Algorithm 4.4 with W
- 3: find an \mathcal{S} -normal path $a_1|\dots|a_p$ equivalent to v' using Algorithm 4.4 with W
- 4: **return** $\bar{b}_q|\dots|\bar{b}_1|a_1|\dots|a_p$

Proposition 5.2. *Assume that \mathcal{C} is a left-Ore category that admits left-lcms, \mathcal{S} is a Garside family in \mathcal{C} , W is a \square -witness on \mathcal{S}^\sharp , and that left-lcms in \mathcal{C} can be computed effectively. Then Algorithm 5.1 returns a symmetric \mathcal{S} -normal decomposition of $[v][u]^{-1}$.*

Proof. By construction, the paths $a_1|\dots|a_p$ and $b_1|\dots|b_q$ are \mathcal{S} -normal, and $b_1\dots b_qg$, which is also $a_1\dots a_pf$, is a left-lcm of f and g . By Lemma 1.20, this implies that $\bar{b}_q|\dots|\bar{b}_1|a_1|\dots|a_p$ is symmetric \mathcal{S} -normal. \square

Example 5.3. Again in the case of the braid monoid B_3^+ as specified by the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{aba} = \mathbf{bab})$ and the Garside family $\mathcal{S} = \{\mathbf{a}, \mathbf{b}, \mathbf{ab}, \mathbf{ba}, \Delta\}$, let us consider the element $g = \mathbf{a}^2\mathbf{ba}^{-1}\mathbf{b}^{-1}$. A decomposition of g as a right-fraction is the signed \mathcal{A} -path $\mathbf{a}|\mathbf{a}|\mathbf{b}|\bar{\mathbf{a}}|\bar{\mathbf{b}}$. By a symmetric argument to that used in Example 2.20, the presentation $(\mathbf{a}, \mathbf{b}; \mathbf{aba} = \mathbf{bab})$ is left-complemented and left-reversing is complete for it. Hence left-reversing computes left-lcms. Here, left-reversing $\mathbf{a}|\mathbf{a}|\mathbf{b}|\bar{\mathbf{a}}|\bar{\mathbf{b}}$ leads to $\bar{\mathbf{b}}|\bar{\mathbf{a}}|\mathbf{b}|\mathbf{a}|\mathbf{a}$ (see Figure 10, left), so $\mathbf{ab} \cdot \mathbf{a}^2\mathbf{b}$, which is also $\mathbf{ba}^2 \cdot \mathbf{ba}$, is a left-lcm of $\mathbf{a}^2\mathbf{b}$ and \mathbf{ba} . The \mathcal{S} -normal decompositions of \mathbf{ab} and \mathbf{ba}^2 respectively are \mathbf{ab} and $\mathbf{ba}|\mathbf{a}$. We conclude that $\bar{\mathbf{ab}}|\mathbf{ba}|\mathbf{a}$ is a symmetric \mathcal{S} -normal decomposition of g .

Algorithm 5.1 is not fully satisfactory in that, as explained in Subsection 4.4, there exists no general solution to the question of computing left-lcms. It may be that the ambient category has been specified by a left-complemented presentation $(\mathcal{A}; \mathcal{R})$ for which left-reversing is complete, in which case left-reversing effectively computes left-lcms, but this need not be the case in general. Now, by (the counterpart of) Proposition 4.22, the natural context for computing left-lcms effectively is when a short left-lcm selector exists.

Lemma 5.4. *Assume that \mathcal{S} is a Garside family in a left-Ore category \mathcal{C} that admits left-lcms. Then there exists a short left-lcm selector on \mathcal{S}^\sharp if and only if \mathcal{S} is strong.*

Proof. For \mathcal{S} to be strong means that, for all a, b with the same target in \mathcal{S}^\sharp , there exist a', b' in \mathcal{S}^\sharp satisfying $a'b = b'a$ and such that a' and b' are left-disjoint. By Lemma 1.20, $a'b$ is a left-lcm of a and b . \square

We are thus led to restate Algorithm 5.1 in the case of a strong Garside family.

Algorithm 5.5 (Symmetric normal decomposition, positive–negative input).

Context: A strong Garside family \mathcal{S} in a left-Ore category \mathcal{C} that admits left-lcms, a \square -witness W and a short lcm-selector $\tilde{\theta}$ on \mathcal{S}^\sharp

Input: A positive–negative \mathcal{S}^\sharp -path $v|\bar{u}$

Output: A symmetric \mathcal{S} -normal decomposition of the element $[v][u]^{-1}$ of $\text{Env}(\mathcal{C})$

- 1: left- $\tilde{\theta}$ -reverse $v|\bar{u}$ into a negative–positive path $\bar{u}'|v'$
- 2: find an \mathcal{S} -normal path $b_1|\dots|b_q$ equivalent to u' using Algorithm 4.4 with W
- 3: find an \mathcal{S} -normal path $a_1|\dots|a_p$ equivalent to v' using Algorithm 4.4 with W
- 4: **return** $\bar{b}_q|\dots|\bar{b}_1|a_1|\dots|a_p$

Proposition 5.6. *Assume that \mathcal{S} is a strong Garside family in a left-Ore category \mathcal{C} that admits left-lcms, W is a \square -witness on \mathcal{S}^\sharp , and $\tilde{\theta}$ is a short left-lcm selector on \mathcal{S}^\sharp . Then Algorithm 5.5 running on a positive–negative \mathcal{S}^\sharp -path $v|\bar{u}$ returns a symmetric \mathcal{S} -normal decomposition of $[v][u]^{-1}$ in time $O((\|u\| + \|v\|)^2)$.*

We skip the proof, which is similar to that of Proposition 5.2.

Example 5.7. With the notation of Example 5.3, the difference is that, in Algorithm 5.5, we use the left-reversing associated with the presentation $(\mathcal{S}; \tilde{\mathcal{R}})$ deduced from $\tilde{\theta}$ rather than the initial presentation $(\mathcal{A}; \mathcal{R})$. The result of left-reversing $a|a|b|\bar{a}|\bar{b}$ is now the length 3 word $\bar{a}\bar{b}|\bar{b}a|a$ (see Figure 10, right), which is equivalent to $\bar{b}|\bar{a}|b|a|a$ as obtained previously (see Figure 10, left). As ab and $ba|a$ are \mathcal{S} -normal, the subsequent normalization steps change nothing, and we conclude again that $\bar{a}\bar{b}|\bar{b}a|a$ is a symmetric \mathcal{S} -normal decomposition of g .

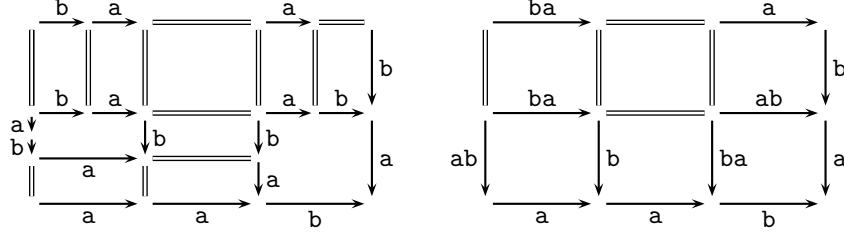
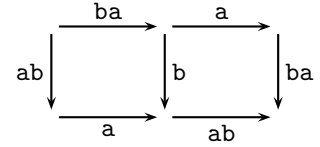


FIGURE 10. Left-reversing of the word $a|a|b|\bar{a}|\bar{b}$ using the left-complement on $\{a, b\}$ (left) and the left-lcm selector on the Garside family $\{a, b, ab, ba, \Delta\}$ (right). Here the right diagram corresponds to gathering pieces of the left diagram, but this would not necessarily be the case for more complicated examples.

Remark 5.8. Algorithm 5.5 (as well as Algorithm 5.1) obeys the scheme “left-reverse, then normalize”. We could as well use switch the operations into “normalize, then left-reverse”, as suggested in Figure 11. The property that guarantees the correctness of the switched version is the third domino rule (Lemma 1.25): left-reversing \mathcal{S} -normal paths always produces \mathcal{S} -normal paths.

For instance, in the context of Example 5.7, we could first find \mathcal{S} -normal words equivalent to $a|a|b$ and $b|a$, namely $a|ab$ and ba , and then left-reverse the (length 3) signed word $a|ab|\bar{b}a$. Of course, the result is $\bar{a}\bar{b}|\bar{b}a|a$ once more, but the involved grid is now the one given on the right.



Starting from an arbitrary path. Extending Algorithm 5.5 to start with an arbitrary path rather than with a positive–negative path is easy. Indeed, we saw that every Garside family provides a short right-lcm selector and using the associated right-reversing transforms an arbitrary path into an equivalent positive–negative path, if some exists.

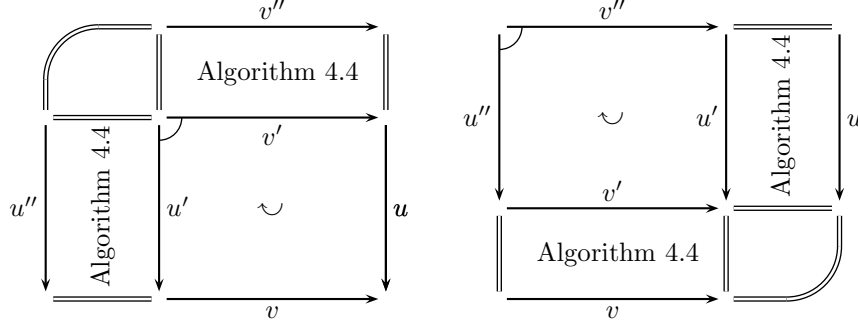


FIGURE 11. Switching normalization and left-reversing in Algorithm 5.5: on the left, we first left-reverse $v|\bar{u}$ into $\bar{u}'|v'$ and then normalize u' into u'' and v' into v'' ; on the right, we first normalize u into u' and v into v' and then left-reverse $v'|\bar{u}'$ into $\bar{u}''|v''$.

Algorithm 5.9 (Symmetric normal decomposition, general input).

Context: A strong Garside family \mathcal{S} in a left-Ore category \mathcal{C} that admits left-lcms, a \square -witness W , a short right-lcm selector θ on \mathcal{S}^\sharp , a short left-lcm selector $\tilde{\theta}$ on \mathcal{S}^\sharp

Input: A signed \mathcal{S}^\sharp -path w

Output: A symmetric \mathcal{S}^\sharp -normal decomposition of $[w]$

- 1: $O :=$ the return value of Algorithm 2.5 with θ running on w
- 2: **if** O is a positive–negative path $v|\bar{u}$ **then**
- 3: compute a symmetric \mathcal{S} -normal decomposition $\bar{u}'|v'$ of $u|\bar{v}$ (Algorithm 5.5)
- 4: **return** $\bar{u}'|v'$
- 5: **else** $\triangleright O$ is fail
- 6: **return fail** $\triangleright [w]$ has no symmetric \mathcal{S} -normal decomposition

Proposition 5.10. *Assume that \mathcal{S} is a strong Garside family in an Ore category \mathcal{C} , W is a \square -witness, θ is a short right-lcm selector on \mathcal{S}^\sharp , and $\tilde{\theta}$ is a short left-lcm selector on \mathcal{S}^\sharp . Then Algorithm 5.9 running on a signed \mathcal{S}^\sharp -path w returns a symmetric \mathcal{S} -normal decomposition of $[w]$ in time $O(\|w\|^2)$.*

Proof. By Proposition 5.6, the final sequence $\bar{u}'|v'$ is a symmetric \mathcal{S} -normal decomposition of the element $[v|\bar{u}]$ of $\text{Env}(\mathcal{C})$. Now, by construction, $v|\bar{u}$ represents the same element of $\text{Env}(\mathcal{C})$ as the initial path w .

The complexity follows from Lemma 2.9 and Propositions 4.5 and 5.6. \square

Example 5.11. Again in the B_3^+ context, let w be the signed word $w = \bar{b}|\bar{a}|b|a|a$. Right-reversing w using the right-lcm selector on $\{1, a, b, ab, ba, \Delta\}$ yields the positive–negative word $a|ab|\bar{b}a$. Left-reversing the latter using the left-lcm selector yields $\bar{a}b|ba|a$. Normalizing ab respectively $ba|a$ do not change these words, and we conclude (once again) that w is symmetric normal.

Remark 5.12. The Baumslag–Solitar presentation $(a, b; a = b^2ab)$ gives an example of a monoid that is left-Ore with left-lcms but not Ore. For instance, as follows from Example 2.8, the elements a and ba admit no common right-multiple. As a consequence, the element $a^{-1}ba$ of the group $\langle a, b \mid a = b^2ab \rangle$ admits no expression as a right-fraction, and no expression as a left-fraction $f^{-1}g$ where f, g are left-disjoint.

Incremental method. Besides Algorithms 5.5 and 5.9, which are global, we can also look for local methods in the spirit of Algorithms 4.2 and 4.4, that is, methods for computing, say, a symmetric normal decomposition of cg from a symmetric normal decomposition of g when c lies in the reference Garside family. This is easy.

Algorithm 5.13 (Left-multiplication—see Figure 12).

Context: A strong Garside family \mathcal{S} in a left-Ore category \mathcal{C} that admits left-lcms, a \square -witness W , a short left-lcm selector $\tilde{\theta}$ on \mathcal{S}^\sharp

Input: A symmetric \mathcal{S} -normal decomposition $\overline{b}_q | \cdots | \overline{b}_1 | a_1 | \cdots | a_p$ of an element g of $\text{Env}(\mathcal{C})$ and an element c of \mathcal{S}^\sharp

Output: A symmetric \mathcal{S} -normal decomposition of bg , if the latter is defined

- 1: $c_{-q} := c$
- 2: **for** i decreasing from q to 1 **do**
- 3: $b'_i := \tilde{\theta}(c_{-i}, b_i); c_{-i+1} := \tilde{\theta}(b_i, c_{-i})$
- 4: **for** i increasing from 1 to p **do**
- 5: $(a'_i, c_i) := W(c_{i-1}, a_i)$
- 6: **return** $\overline{b'_q} | \cdots | \overline{b'_1} | a'_1 | \cdots | a'_p | c_p$

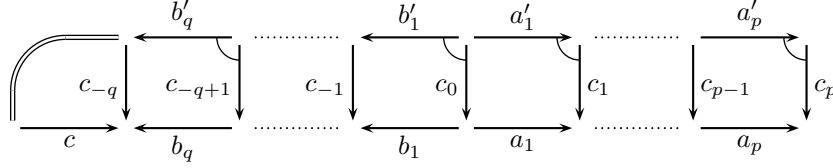


FIGURE 12. Algorithm 5.13: starting from c in \mathcal{S}^\sharp and a symmetric \mathcal{S} -normal decomposition $\overline{b}_q | \cdots | \overline{b}_1 | a_1 | \cdots | a_p$ of g , one obtains a symmetric \mathcal{S} -normal decomposition of cg .

Proposition 5.14. *Assume that \mathcal{S} is a strong Garside family in a left-Ore category \mathcal{C} that admits left-lcms, W is a \square -witness on \mathcal{S}^\sharp , and $\tilde{\theta}$ is a short left-lcm selector on \mathcal{S}^\sharp . Then Algorithm 5.13 running on a symmetric \mathcal{S} -normal decomposition of an element g of $\text{Env}(\mathcal{C})$ and c in \mathcal{S}^\sharp returns a symmetric \mathcal{S} -normal decomposition of cg if the latter is defined. The map $\tilde{\theta}$ is invoked $2q$ times and the map W is invoked p times.*

Proof. The third domino rule (Lemma 1.25) implies that $b'_1 | \cdots | b'_q$ is \mathcal{S} -greedy. The fourth domino rule implies that b'_1 and a'_1 are left-disjoint. Next, the first domino rule implies that $a'_1 | \cdots | a'_p$ is \mathcal{S} -normal. Finally, $a'_p | c_p$ is \mathcal{S} -normal by construction. Hence $\overline{b'_q} | \cdots | \overline{b'_1} | a'_1 | \cdots | a'_p | c_p$ is symmetric \mathcal{S} -normal. The commutativity of the diagram in Figure 12 then guarantees that the latter sequence is a decomposition of $cb_q^{-1} \cdots b_1^{-1} a_1 \cdots a_p$, hence of cg . \square

The symmetry of the definition of symmetric normal paths implies that there exists a counterpart of Algorithm 5.13 computing a symmetric normal decomposition of gc^{-1} from one of g for c in the reference Garside family \mathcal{S} . By contrast, there exists no simple general method for left-division (or right-multiplication). However, when \mathcal{S} happens to be bounded, such a method exists, because left-dividing by an element $\Delta(x)$ is easy. Now, if \mathcal{S} is bounded by Δ and a belongs to $\mathcal{S}(x, -)$, then $a\partial(a) = \Delta(x)$ holds, so that left-dividing by a can be decomposed into left-dividing by $\Delta(x)$ and then left-multiplying by $\partial(a)$. Hereafter, in order to simplify notation, we shall often skip the source of the considered Δ -element, thus writing $\Delta(-)$ instead of $\Delta(x)$ when mentioning x explicitly is not necessary.

Algorithm 5.15 (Left-division—see Figure 13).

Context: A Garside family \mathcal{S} bounded by a map Δ in an Ore category \mathcal{C} , a \square -witness W for \mathcal{S}^\sharp

Input: A symmetric \mathcal{S} -normal decomposition $\overline{b}_q | \cdots | \overline{b}_1 | a_1 | \cdots | a_p$ of g in $\text{Env}(\mathcal{C})$, an element c of \mathcal{S}^\sharp

Output: A symmetric \mathcal{S} -normal decomposition of $c^{-1}g$, if the latter is defined

- 1: $c_{-q} := c$
- 2: **for** i decreasing from q to 1 **do**
- 3: $(c_{-i+1}, b'_i) := W(b_i, c_{-i})$
- 4: $c'_0 := \tilde{\partial}(c_0)$
- 5: **for** i increasing from 1 to p **do**
- 6: $(a'_i, c'_i) := W(c'_{i-1}, a_i)$
- 7: $b'_0 := \partial(a'_1)$
- 8: **return** $\overline{b'_q} | \cdots | \overline{b'_1} | b'_0 | a'_2 | \cdots | a'_p | c'_p$

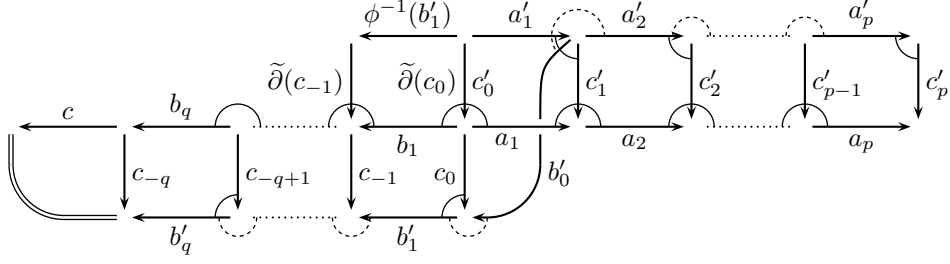


FIGURE 13. Left-division by an element of \mathcal{S}^\sharp starting from a symmetric \mathcal{S} -normal path $\overline{b_q} | \dots | \overline{b_1} | a_1 | \dots | a_p$.

Proposition 5.16. *Assume that \mathcal{S} is a Garside family that is bounded by a map Δ in an Ore category \mathcal{C} . Then Algorithm 5.15 running on a symmetric \mathcal{S} -normal decomposition of an element g of $\text{Env}(\mathcal{C})$ and c in \mathcal{S}^\sharp returns a symmetric \mathcal{S} -normal decomposition of $c^{-1}g$ if the latter is defined.*

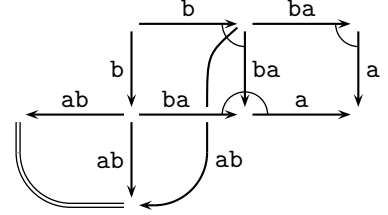
Proof. By construction, the diagram of Figure 13 is commutative, so the returned path is equivalent to $\overline{c} | \overline{b_q} | \dots | \overline{b_1} | a_1 | \dots | a_p$, hence it is a decomposition of $c^{-1}g$, and its entries are elements of \mathcal{S}^\sharp . So the point is to check that the path is symmetric \mathcal{S} -greedy.

First, as $b_i | b_{i+1}$ is \mathcal{S} -normal for $i = 1, \dots, q-1$, the first domino rule (Lemma 1.16) implies that $b'_i | b'_{i+1}$ is \mathcal{S} -normal as well. Similarly, as $a_i | a_{i+1}$ is \mathcal{S} -normal for $i = 1, \dots, p-1$, the first domino rule (Lemma 1.16) implies that $a'_i | a'_{i+1}$ is \mathcal{S} -normal as well.

So it remains to check that $b'_0 | b'_1$ is \mathcal{S} -normal and that b'_0 and a'_2 are left-disjoint. Consider $b'_0 | b'_1$. Assume that a is an element of \mathcal{S}^\sharp satisfying $a \preceq \phi^{-1}(b'_1)$ and $a \preceq a'_1$. A fortiori, we have $a \preceq \phi^{-1}(b'_1) \tilde{\partial}(c_{-1})$ and $a \preceq a'_1 c'_1$, that is, $a \preceq c'_0 b_1$ and $a \preceq c'_0 a_1$. By assumption, b_1 and a_1 are left-disjoint, hence we deduce $a \preceq c'_0$. Now, by assumption, $c_0 | b'_1$ is \mathcal{S} -normal, hence, by Lemma 1.30, $\partial(c_0)$ and b'_1 have no nontrivial common left-divisor. As ϕ and, therefore, ϕ^{-1} is an automorphism (Lemma 1.29), it follows that $\phi^{-1}(\partial(c_0))$ and $\phi^{-1}(b'_1)$ have no nontrivial common left-divisor. Now, by definition, $\phi^{-1}(\partial(c_0))$ is c'_0 . As we have $a \preceq \phi^{-1}(b'_1)$ and $a \preceq c'_0$, we deduce that a is invertible and, therefore, $\phi^{-1}(b'_1)$ and a'_1 have no nontrivial common left-divisor. Applying the automorphism ϕ , we deduce that b'_1 and $\phi(a'_1)$, that is, b'_1 and $\partial(b'_0)$, have no nontrivial common left-divisor. By Lemma 1.30 again, this implies that $b'_0 | b'_1$ is \mathcal{S} -normal.

Finally, again by Lemma 1.30, the assumption that $a'_1 | a'_2$ is \mathcal{S} -normal implies that $\partial(a'_1)$ and a'_2 are left-disjoint: by definition, $\partial(a'_1)$ is b'_0 , so b'_0 and a'_2 are left-disjoint. So $\overline{b'_q} | \dots | \overline{b'_1} | b'_0 | a'_2 | \dots | a'_p | c'_p$ is indeed a symmetric \mathcal{S} -normal path. \square

Example 5.17. Consider $(ab)^{-1}(ba^2)$ in B_3^+ once more. Suppose that we found that $ba | a$ is an \mathcal{S} -normal decomposition of ba^2 , and we wish to find a (the) \mathcal{S} -normal decomposition of $(ab)^{-1}(ba^2)$. Then applying Algorithm 5.15 amounts to completing the diagram on the right, in which of course we read the expected symmetric normal decomposition $\overline{ab} | ba | a$.



If, in Algorithm 5.15, c is $\Delta(x)$, then each element c_{-i} is of the form $\Delta(-)$, so that b'_i is simply $\phi(b_i)$, so that c'_0 is trivial, and so are all elements c_i with $i \geq 0$. We immediately deduce

Corollary 5.18. *Assume that \mathcal{S} is a Garside family that is bounded by a map Δ in an Ore category \mathcal{C} . If $\overline{b_q} | \dots | \overline{b_1} | a_1 | \dots | a_p$ is a symmetric \mathcal{S} -normal decomposition of g , then a symmetric \mathcal{S} -normal decomposition of $\Delta(x)^{-1}g$ (where x is the source of g) is $\overline{\phi(b_q)} | \dots | \overline{\phi(b_1)} | \partial(a_1) | a_2 | \dots | a_p$.*

5.2. Computing Δ -normal decompositions. When a Garside family is bounded, alternative distinguished decompositions for the elements of the ambient category and its groupoid of fractions arise, namely the Δ -normal decompositions. We now describe incremental methods for computing such decompositions, namely algorithms that, when starting from a Δ -normal decomposition of an element g and an element c of the considered Garside family, return Δ -normal decompositions of cg and $c^{-1}g$, respectively.

Algorithm 5.19 (Left-multiplication for Δ -normal—see Figure 14).

Context: A Garside family \mathcal{S} that is bounded by a map Δ in an Ore category \mathcal{C} , a \square -witness W for \mathcal{S}^\sharp , an $=^\times$ -test E in \mathcal{C}

Input: A Δ -normal decomposition $\Delta^{[n]}(x)|a_1|\cdots|a_p$ of an element g of $\text{Env}(\mathcal{C})$, an element c of \mathcal{S}^\sharp

Output: A Δ -normal decomposition of cg , if the latter is defined

- 1: $c_0 := \phi^n(c)$
- 2: **for** i increasing from 1 to p **do**
- 3: $(a'_i, c_i) := W(c_{i-1}, a_i)$
- 4: **if** $E(\Delta(-), a'_1) \neq \perp$ **then** $\triangleright a'_1$ is Δ -like
- 5: $a'_2 := E(\Delta(-), a'_1)a'_2$
- 6: **return** $\Delta^{[n+1]}(-)|a'_2|\cdots|a'_p|c_p$
- 7: **else** $\triangleright a'_1$ is not Δ -like
- 8: **return** $\Delta^{[n]}(-)|a'_1|\cdots|a'_p|c_p$

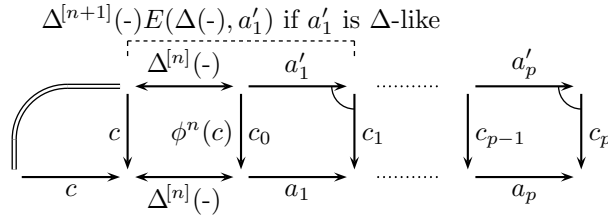


FIGURE 14. Computing a Δ -normal decomposition of cg starting from a Δ -normal decomposition of g : the $|n|$ entries of the type $\Delta(-)$ are treated directly using ϕ^n , and then it just remains to left-multiply a (positive) Δ -normal sequence by an element of \mathcal{S} using the standard method. Note that, depending on the sign of n , the diagrams for $\Delta^{(n)}(-)$ consists either of n right-oriented arrows (case $n \geq 0$) or of $|n|$ left-oriented arrows (case $n \leq 0$).

Algorithm 5.20 (Left-division for Δ -normal—see Figure 15).

Context: A Garside family \mathcal{S} that is bounded by a map Δ in an Ore category \mathcal{C} , a \square -witness W for \mathcal{S}^\sharp , an $=^\times$ -test E in \mathcal{C}

Input: A Δ -normal decomposition $\Delta^{[n]}(x)|a_1|\cdots|a_p$ for an element c of $\text{Env}(\mathcal{C})$ and an element c of \mathcal{S}^\sharp with source x

Output: An Δ -normal decomposition of $c^{-1}g$

- 1: $c_0 := \phi^n(\tilde{\partial}(c))$
- 2: **for** i increasing from 1 to p **do**
- 3: $(a'_i, c_i) := W(c_{i-1}, a_i)$
- 4: **if** $E(\Delta(-), a'_1) \neq \perp$ **then** $\triangleright a'_1$ is Δ -like
- 5: $a'_2 := E(\Delta(-), a'_1)a'_2$
- 6: **return** $\Delta^{[n]}(-)|a'_2|\cdots|a'_p|c_p$
- 7: **else** $\triangleright a'_1$ is not Δ -like
- 8: **return** $\Delta^{[n-1]}(-)|a'_1|\cdots|a'_p|c_p$

Proposition 5.21. *Assume that \mathcal{S} is a Garside family that is bounded by a map Δ in an Ore category \mathcal{C} , W is a \square -witness on \mathcal{S} , and E is an $=^\times$ -test in \mathcal{C} . Then Algorithm 5.19 (resp. Algorithm 5.20) running on a Δ -normal decomposition of an element g of $\text{Env}(\mathcal{C})$ and c in \mathcal{S}^\sharp returns a Δ -normal decomposition of cg (resp. $c^{-1}g$) (when the latter is defined).*

Proof. Consider first Algorithm 5.19. Then the diagram of Figure 14 is commutative as, in particular, $c\Delta^{(n)}(-) = \Delta^{(n)}(u)\phi^n(c)$ holds whenever defined. As usual, the first domino rule implies that $a'_1|\cdots|a'_p$ is \mathcal{S} -normal and, by construction, $a'_p|c_p$ is \mathcal{S} -normal. So it just remains to

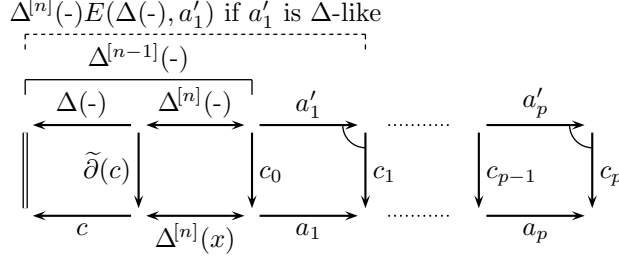
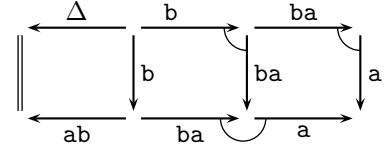


FIGURE 15. Algorithm 5.20 : Computing a Δ -normal decomposition for $c^{-1}g$ from a Δ -normal decomposition of g .

consider the relation between $\Delta^{[n]}(-)$ and a'_1 . Now, if a'_1 is not Δ -like, $\Delta^{[n]}(-)|a'_1$ is Δ -normal. On the other hand, if a'_1 is Δ -like, then, by definition, we have $a'_1 = \Delta(-)E(\Delta(-), a'_1)$, so that $\Delta(-)$ can be incorporated to $\Delta^{[n]}(-)$ to form $\Delta^{[n+1]}(-)$ and $E(\Delta(-), a'_1)$ can be incorporated to a'_2 to form a new element of \mathcal{S}^\sharp as $\mathcal{C}^\times \mathcal{S}^\sharp \subseteq \mathcal{S}^\sharp$ holds. Then $\Delta^{[n+1]}(-)|a'_2$ must be Δ -normal because the new element a'_2 may not be Δ -like. Indeed, $\Delta(-) \preceq a'_2$ would imply $\Delta^{(2)}(-) \preceq a'_1 a'_2$, whence $\Delta^{(2)}(-) \preceq a'_1 a'_2 c_2 = c_0 a_1 a_2$. Now, as c_0 lies in \mathcal{S}^\sharp , we can write $\Delta^{(2)}(-) = c_0 \partial(c_0) \Delta(-) = c_0 \Delta(-) \phi(\partial(c_0))$, so $\Delta^{(2)}(-) \preceq c_0 a_1 a_2$ implies $c_0 \Delta(-) \phi(\partial(c_0)) \preceq c_0 a_1 a_2$, whence $\Delta(-) \preceq a_1 a_2$, contradicting the assumption that $\Delta^{[n]}(-)|a_1$ is Δ -normal. So no cascade may occur here.

The argument for Algorithm 5.20 is entirely similar. The only new point is that, in the diagram of Figure 15, the left square is commutative because $\tilde{\partial}(c)c = \Delta(-)$ holds for every c in \mathcal{S}^\sharp . \square

Example 5.22. As in Example 5.17, assume that we know that $\mathbf{ba|a}$ is a Δ -normal decomposition of $\mathbf{b}^2\mathbf{a}$ and we wish to find a (the) Δ -normal decomposition of $(\mathbf{ab})^{-1}(\mathbf{b}^2\mathbf{a})$. Applying Algorithm 5.20 amounts to completing the diagram on the right, and we read that the expected decomposition is $\Delta^{[-1]}|\mathbf{b|ba|a}$.



5.3. Solving the Word Problem. As in the positive case, we shall describe several solutions for the Word Problem in the involved groupoid of fractions

Using symmetric normal decompositions. First, symmetric normal decompositions give an direct solution extending Algorithm 4.12 to the signed case.

Algorithm 5.23 (Word Problem, general case I).

Context: A strong Garside family \mathcal{S} in an Ore category \mathcal{C} that admits left-lcms, a \square -witness W , a short left-lcm selector $\tilde{\theta}$, and an $=^\times$ -test E on \mathcal{S}^\sharp

Input: A signed \mathcal{S}^\sharp -path w

Output: **true** if w represents an element 1_x in $\text{Env}(\mathcal{C})$, and **false** otherwise

- 1: $x := \text{src}(w)$; $y := \text{trg}(w)$
- 2: **if** $x \neq y$ **then**
- 3: **return false**
- 4: **else**
- 5: use Algorithm 5.9 to find a symmetric \mathcal{S} -normal path $\bar{u}|v$ that represents $[w]$
- 6: **return** the value of $\text{COMPARENORMALPATHS}(u, v)$ (Algorithm 4.12)

Proposition 5.24. *Assume that \mathcal{S} is a strong Garside family \mathcal{S} in an Ore category \mathcal{C} that admits left-lcms, W is a \square -witness on \mathcal{S} , $\tilde{\theta}$ is a short left-lcm selector on \mathcal{S}^\sharp , and E is an $=^\times$ -test on \mathcal{S}^\sharp . Then Algorithm 5.23 running on a signed \mathcal{S}^\sharp -path w of length at most ℓ decides in time $O(\ell^2)$ whether $[w]$ is an identity-element in $\text{Env}(\mathcal{C})$.*

Proof. By Proposition 5.10, line 5 of Algorithm 5.23 computes a symmetric \mathcal{S} -normal path $\bar{u}|v$ representing $[w]$ in time $O(\ell^2)$. The signed path w represents an identity-element in $\text{Env}(\mathcal{C})$ if

and only if the (positive) paths u and v represent the same element in \mathcal{C} . The claim then follows with Proposition 4.13. \square

Using Δ -normal decompositions. According to Proposition 1.35, when they exist, Δ -normal decompositions enjoy the same uniqueness property as symmetric normal decompositions. Therefore, in the case when the reference Garside family is not only strong but even bounded, Algorithm 5.23 and Proposition 5.24 admit exact counterparts where symmetric normal is replaced with Δ -normal. We skip the details.

Using reversing. As in the positive case, at least when the ambient category contains no nontrivial invertible element and admits lcms, we can also solve the Word Problem by using reversing and thus avoiding to compute distinguished decompositions. To stick to the context described in Subsection 2.1 we assume here that the ambient category admits left- and right-lcms and contains no nontrivial invertible element, which amounts to requiring that the lcms are unique.

Algorithm 5.25 (Word Problem, general case II).

Context: A strong Garside family \mathcal{S} in an Ore category \mathcal{C} that admits unique right- and left-lcms

Input: A signed \mathcal{S} -path w

Output: **true** if w represents an element 1_x in $\text{Env}(\mathcal{C})$, and **false** otherwise

- 1: $\theta :=$ the right-lcm selector on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$
- 2: $\tilde{\theta} :=$ the left-lcm selector on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$
- 3: right- θ -reverse w into a positive–negative path $u|\bar{v}$ using Algorithm 2.5
- 4: left- $\tilde{\theta}$ -reverse $u|\bar{v}$ into a negative–positive path $\bar{u}'|v'$ using (the left counterpart of) Algorithm 2.3
- 5: **return** $\llbracket u' = v' = \varepsilon_- \rrbracket$

Lemma 2.9, Proposition 1.15, and their left counterparts imply:

Proposition 5.26. *Assume that \mathcal{S} is a strong Garside family \mathcal{S} in an Ore category \mathcal{C} that admits unique right- and left-lcms.*

- (i) *Algorithm 5.25 solves the Word Problem of $\text{Env}(\mathcal{C})$ with respect to \mathcal{S} .*
- (ii) *If \mathcal{S} is finite, the complexity of Algorithm 5.25 is quadratic in the length of the input path.*

Appealing to a right- and a left-reversing, Algorithm 5.25 is nicely symmetric. However, it requires the existence of right-lcms, which is not guaranteed in every Ore category that admits a strong Garside family. Actually, this assumption is superfluous, since a double left-reversing can be used instead.

Algorithm 5.27 (Word Problem, general case III).

Context: A strong Garside family \mathcal{S} in an Ore category \mathcal{C} that admits unique left-lcms

Input: A signed \mathcal{S} -path w

Output: **true** if w represent an element 1_x in $\text{Env}(\mathcal{C})$, and **false** otherwise

- 1: $\tilde{\theta} :=$ the left-lcm selector on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$
- 2: left- $\tilde{\theta}$ -reverse w into a negative–positive $\bar{u}|v$ using (the left counterpart of) Algorithm 2.5
- 3: left- $\tilde{\theta}$ -reverse $v|\bar{u}$ into a negative–positive path $\bar{u}'|v'$ using (the left counterpart of) Algorithm 2.3
- 4: **return** $\llbracket u' = v' = \varepsilon_- \rrbracket$

Proposition 5.28. *Assume that \mathcal{S} is a strong Garside family in an Ore category \mathcal{C} that admits unique left-lcms.*

- (i) *Algorithm 5.27 solves the Word Problem of $\text{Env}(\mathcal{C})$ with respect to \mathcal{S} .*
- (ii) *If \mathcal{S} is finite, the complexity of Algorithm 5.27 is quadratic in the length of the input path.*

Proof. The signed path w represents an identity-element in $\text{Env}(\mathcal{C})$ if and only if the paths u and v represent the same element in \mathcal{C} . By the counterpart of Proposition 1.15, left- $\tilde{\theta}$ -reversing is

complete and terminating, so u and v represent the same element of \mathcal{C} if and only if $v|\bar{u}$ is left- $\tilde{\theta}$ -reversible to an empty path. By Lemma 2.9, the complexity of Algorithm 5.27 is in $O(\|w\|^2)$. \square

Example 5.29. Running on the signed word $w = \bar{a}\bar{b}|ba|a$, Algorithm 5.25 right-reverses w into $a|ab|\bar{b}\bar{a}$, and then left-reverses the latter words back to $\bar{a}\bar{b}|ba|a$. The final word is not empty, hence w does not represent 1 in the group B_3 .

Running on w , Algorithm 5.27 first left-reverses w into itself (since w is a negative–positive word), then switches the numerator and the denominator into $ba|a|\bar{a}\bar{b}$, and finally left-reverses the latter word. One finds now $\bar{a}|b|b$, a nonempty word, and one concludes again that w does not represent 1.

Remark 5.30. As in Section 4, we could also state a result referring to an arbitrary generating family \mathcal{A} that satisfies convenient properties but that is not necessarily a strong Garside family.

5.4. Computing decompositions for an inverse. We complete the analysis of our distinguished decompositions with explicit methods for finding a decomposition of g^{-1} from one of g .

In the case of symmetric normal decompositions, the result is trivial:

Proposition 5.31. *Assume that \mathcal{C} is a left-Ore category and \mathcal{S} is a Garside family in \mathcal{C} . If $\bar{b}_q|\dots|\bar{b}_1|a_1|\dots|a_p$ is a symmetric \mathcal{S} -normal decomposition for an element g of $\text{Env}(\mathcal{C})$, then $\bar{a}_p|\dots|\bar{a}_1|b_1|\dots|b_q$ is a symmetric \mathcal{S} -normal decomposition for g^{-1} .*

Proof. The definition makes it obvious that $\bar{a}_p|\dots|\bar{a}_1|b_1|\dots|b_q$ is symmetric \mathcal{S} -normal whenever $\bar{b}_q|\dots|\bar{b}_1|a_1|\dots|a_p$ is. On the other hand, $g = b_q^{-1}\dots b_1^{-1}a_1\dots a_p$ implies $g^{-1} = a_p^{-1}\dots a_1^{-1}b_1\dots b_q$ in $\text{Env}(\mathcal{C})$. \square

The case of Δ -normal decompositions is more complicated, as the definition is not symmetric. However, finding an explicit formula is not difficult.

Proposition 5.32. *Assume that \mathcal{S} is a Garside family that is bounded by Δ in a cancellative category \mathcal{C} . If $\Delta^{(n)}|a_1|\dots|a_p$ is a Δ -normal decomposition for an element g of $\text{Env}(\mathcal{C})$, then $\Delta^{(-n-p)}|\partial(\phi^{-n-p}(a_p))|\dots|\partial(\phi^{-n-1}(a_1))$ is a Δ -normal decomposition for g^{-1} .*

Proof. We first check that the sequence mentioned in the statement is Δ -normal. So let $i < p$. By hypothesis, $a_i|a_{i+1}$ is \mathcal{S} -normal. As ϕ is an automorphism of \mathcal{S} , the path $\phi^{-n-i}(a_i)|\phi^{-n-i}(a_{i+1})$ is also \mathcal{S} -normal. By Lemma 1.30, this implies that $\partial(\phi^{-n-i}(a_i))$ and $\phi^{-n-i}(a_{i+1})$ are left-disjoint. The latter element is $\phi(\phi^{-n-i-1}(a_{i+1}))$, hence it is $\partial(\partial(\phi^{-n-i-1}(a_{i+1})))$. Reading the above disjointness result in a symmetric way and applying Lemma 1.30 again, we deduce that $\partial(\phi^{-n-i-1}(a_{i+1}))|\partial(\phi^{-n-i}(a_i))$ is \mathcal{S} -normal. Hence $\partial(\phi^{-n-p}(a_p))|\dots|\partial(\phi^{-n-1}(a_1))$ is \mathcal{S} -normal.

Moreover, the hypothesis that a_1 is not Δ -like implies that $\phi^{-n-1}(a_1)$ is not Δ -like either, and, therefore, $\partial(\phi^{-n-1}(a_1))$ is not invertible. On the other hand, the hypothesis that a_p is not invertible implies that $\phi^{-n-p}(a_p)$ is not invertible either, and, therefore, $\partial(\phi^{-n-p}(a_p))$ is not Δ -like. Hence, we conclude that the path $\Delta^{(-n-p)}|\partial(\phi^{-n-p}(a_p))|\dots|\partial(\phi^{-n-1}(a_1))$ is Δ -normal.

It remains to check that the latter Δ -normal path is a decomposition of g^{-1} . Now, by construction, the maps ϕ and ∂ commute and, pushing the factor $\Delta^{(n)}$ to the left, we obtain the equality

$$(5.33) \quad \Delta^{(-n-p)}(-)\partial(\phi^{-n-p}(a_p))\dots\partial(\phi^{-n-1}(a_1))\cdot\Delta^{(n)}(-)a_1\dots a_p \\ = \Delta^{(-n-p)}(-)\Delta^{(n)}(-)\partial(\phi^{-p}(a_p))\dots\partial(\phi^{-1}(a_1))a_1\dots a_p,$$

and our goal is to prove that this element is 1_y . Let us call the right term of (5.33) $E(n, a_1, \dots, a_p)$. We shall prove using induction on $p \geq 0$ that an expression of the form $E(n, a_1, \dots, a_p)$ equals 1_y , where y is the target of a_p .

Assume first $p = 0$. What remains for $E(n)$ is then $\Delta^{(-n)}(-)\Delta^{(n)}(-)$, which, by the remark following Notation 1.32, is 1_y . Assume $p \geq 1$. The above result enables us to gather the first two entries of E into $\Delta^{(-p)}(-)$. Then $\partial(\phi^{-1}(a_1))a_1$, which is also $\tilde{\partial}(a_1)a_1$, equals $\Delta(z)$, where z is the

source of $\tilde{\partial}(a_1)$. But, then, we can push this $\Delta(-)$ -factor to the left through the $\partial(\phi^{-i}(a_i))$ factors with the effect of diminishing the exponents of ϕ by one. In this way, $E(n, a_1, \dots, a_p)$ becomes

$$\Delta^{(-p)}(-) \Delta(-) \partial(\phi^{-p+1}(a_p)) \dots \partial(\phi^{-1}(a_2)) a_2 \dots a_p,$$

which is $E(1, a_2, \dots, a_p)$. By the induction hypothesis, its value is 1_y . \square

5.5. Computing upper and lower bounds. If \mathcal{C} is an Ore category, the left-divisibility relation of \mathcal{C} naturally extends into a relation on $\text{Env}(\mathcal{C})$, namely the relation, still denoted by \preceq , such that $f \preceq g$ holds if there exists h in \mathcal{C} satisfying $fh = g$. This relation is a partial preordering on $\text{Env}(\mathcal{C})$, and it is a partial ordering if \mathcal{C} admits no nontrivial invertible element. It directly follows from the definition that, if f, g are elements of \mathcal{C} , then a least common upper bound (*resp.* a greatest common lower bound) of f and g with respect to \preceq is (when it exists) a right-lcm (*resp.* a left-gcd) of f and g .

Lemma 5.34. *Assume that \mathcal{C} is an Ore category. Then any two elements of $\text{Env}(\mathcal{C})$ with the same source admit a least common upper bound (*resp.* a greatest common lower bound) with respect to \preceq if and only if any two elements of \mathcal{C} with the same source admit a right-lcm (*resp.* a left-gcd).*

We skip the easy proof. Then a natural computation problem arises whenever the above bounds exist. The solution is easy.

Algorithm 5.35 (Least common upper bound with respect to \preceq).

Context: A strong Garside family \mathcal{S} in an Ore category \mathcal{C} that admits unique left- and right-lcms

Input: Two signed \mathcal{S} -paths w_1, w_2 with the same source

Output: A signed \mathcal{S} -path representing the least common upper bound of $[w_1]$ and $[w_2]$

- 1: $\theta :=$ the right-lcm selector on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$
- 2: $\tilde{\theta} :=$ the left-lcm witness on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$
- 3: left- $\tilde{\theta}$ -reverse $\overline{w_1}|w_2$ into a negative–positive path $\overline{u}|v$ using (the left counterpart of) Algorithm 2.5
- 4: right- θ -reverse $\overline{u}|v$ into a positive–negative path $v'|\overline{u'}$ using Algorithm 2.3
- 5: **return** $w_1|v'$

The solution for greatest lower bound is entirely symmetric.

Algorithm 5.36 (Greatest lower bound with respect to \preceq).

Context: A strong Garside family \mathcal{S} in an Ore category \mathcal{C} that admits unique left- and right-lcms

Input: Two signed \mathcal{S} -paths w_1, w_2 with the same source

Output: A signed \mathcal{S} -path representing the greatest lower bound of $[w_1]$ and $[w_2]$

- 1: $\theta :=$ the right-lcm selector on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$
- 2: $\tilde{\theta} :=$ the left-lcm selector on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$
- 3: right- θ -reverse $\overline{w_1}|w_2$ into a positive–negative path $u|\overline{v}$ using Algorithm 2.5
- 4: left- $\tilde{\theta}$ -reverse $u|\overline{v}$ into a negative–positive path $\overline{v'}|u'$ using (the left counterpart of) Algorithm 2.3
- 5: **return** $w_1|\overline{v'}$

Proposition 5.37. *Assume that \mathcal{S} is a strong Garside family in an Ore category \mathcal{C} that admits unique left- and right-lcms. Then Algorithm 5.35 (*resp.* Algorithm 5.36) running on two signed \mathcal{S} -paths w_1 and w_2 of length at most ℓ returns the least upper bound (*resp.* the greatest lower bound) of $[w_1]$ and $[w_2]$ with respect to \preceq in time $O(\ell^2)$.*

Proof. First consider Algorithm 5.35. By the left counterpart of Lemma 2.9, the left- $\tilde{\theta}$ -reversing in line 3 takes time $O(\ell^2)$ and produces two $\mathcal{S}^\#$ -paths u, v of length $O(\ell)$ such that, in $\text{Env}(\mathcal{C})$, we have $\overline{u}|v = \overline{w_1}|w_2$, hence $[w_1] = g[u]$ and $[w_2] = g[v]$ for some g , namely the common class of the signed paths $w_1|\overline{u}$ and $w_2|\overline{v}$. By Proposition 4.22, the right- θ -reversing in line 4 takes time

$O(\ell^2)$ and produces two \mathcal{S} -paths u', v' such that $u|v'$ and $v|u'$ both represent the right-lcm of $[u]$ and $[v]$, hence their least upper bound with respect to \preceq . As the partial ordering \preceq is invariant under left-multiplication, the claim follows.

The claim for Algorithm 5.36 follows in an analogous way from Lemma 2.9 and the left counterpart of Proposition 4.22, noting that, for all f, g in $\mathcal{E}nv(\mathcal{C})$ and h in \mathcal{C} , the conditions $fh = g$ and $f^{-1} = hg^{-1}$ are equivalent. \square

When we apply Algorithms 5.35 and 5.36 to a pair of positive paths, we obtain algorithms that determine right-lcms and left-gcds. In this case, Algorithm 5.35 is simply Algorithm 2.3, since the left-reversing step is trivial: the initial path $\overline{w_1}|w_2$ is directly negative–positive. Algorithm 5.36, however, has no equivalent in Section 4. Its output is a path representing the left-gcd but, in general, it need not be a positive path, although it must be equivalent to a positive path. In order to obtain a positive output, a third reversing step can be added.

Algorithm 5.38 (Left-gcd).

Context: A strong Garside family \mathcal{S} in an Ore category \mathcal{C} that admits unique left- and right-lcms

Input: Two \mathcal{S} -paths u, v with the same source

Output: An \mathcal{S} -path that represents the left-gcd of $[u]$ and $[v]$ in \mathcal{C}

- 1: $\theta :=$ the right-lcm selector on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$
- 2: $\tilde{\theta} :=$ the left-lcm selector on $\mathcal{S} \cup \mathbf{1}_{\mathcal{C}}$
- 3: right- θ -reverse $\overline{u}|v$ into a positive–negative path $v'|\overline{u'}$ using Algorithm 2.3
- 4: left- $\tilde{\theta}$ -reverse $v'|\overline{u'}$ into a negative–positive path $\overline{u''}|v''$ using (the left counterpart of) Algorithm 2.3
- 5: left- $\tilde{\theta}$ -reverse $u|\overline{u''}$ into a positive path w using (the left counterpart of) Algorithm 2.3
- 6: **return** w

Proposition 5.39. *Assume that \mathcal{S} is a strong Garside family in an Ore-category that admits unique left- and right-lcms. Then Algorithm 5.38 running on two \mathcal{S} -paths u, v of length at most ℓ returns the left-gcd of $[u]$ and $[v]$ in time $O(\ell^2)$.*

Proof. It follows from Proposition 5.37 that, for u'' as computed in line 4, the path $u|\overline{u''}$ represents the left-gcd of $[u]$ and $[v]$. As u and v are both positive, this path $u|\overline{u''}$ must be equivalent to some positive path, say u_0 . Then the signed path $u|\overline{u''}|\overline{u_0}$ represents an identity-element in $\mathcal{E}nv(\mathcal{C})$, so the positive paths u and u_0u'' are equivalent. Since, by the counterpart of Proposition 1.15, left-reversing is complete for the considered presentation, the path $u|\overline{u''}|\overline{u_0}$ is left-reversible to an empty path, that is, the left-reversing grid constructed from $u|\overline{u''}|\overline{u_0}$ has empty arrows everywhere on the left and on the top. This implies in particular that the subgrid corresponding to left-reversing $u|\overline{u''}$ has empty arrows on the left, that is, with our current notation, that the word w computed at line 5 is positive (and equivalent to u_0).

By Proposition 5.37, the computation time of w lies in $O(\ell^2)$ since, by the left counterpart of Lemma 2.4, the left- $\tilde{\theta}$ -reversing in line 5 takes time $O(\ell^2)$. \square

Example 5.40. Let us consider a last time the monoid B_3^+ and the elements represented by $u = \mathbf{a|b|b}$ and $v = \mathbf{b|a|b|b}$. As seen in Figure 9, right-reversing $\overline{u}|v$ using the right-lcm selector leads to the positive–negative word $\mathbf{ab|\bar{a}}$. Then left-reversing the latter word using the left-lcm selector leads to the negative–positive word $\mathbf{\bar{b}|ab}$. Finally, left-reversing $\mathbf{a|b|b|\bar{b}}$ gives the positive word $\mathbf{a|b}$, and we conclude that the left-gcd of $[u]$ and $[v]$ in B_3^+ is \mathbf{ab} . Note that, instead of using \mathcal{S} -words and the associated lcm selector, we could instead use \mathcal{A} -words and the initial right-complement associated with the Artin presentation: the uniqueness of the final result guarantees that the successive involves words must be pairwise equivalent.

REFERENCES

- [1] S.I. Adyan, *Fragments of the word Delta in a braid group*, Mat. Zam. Acad. Sci. SSSR **36-1** (1984) 25–34; translated Math. Notes of the Acad. Sci. USSR; 36-1 (1984) 505–510.
- [2] E. Artin, *Theory of Braids*, Ann. of Math. **48** (1947) 101–126.
- [3] E. Brieskorn & K. Saito, *Artin-Gruppen und Coxeter-Gruppen*, Invent. Math. **17** (1972) 245–271.
- [4] R. Charney, *Artin groups of finite type are biautomatic*, Math. Ann. **292-4** (1992) 671–683.
- [5] A.H. Clifford & G.B. Preston, *The algebraic Theory of Semigroups, vol. 1*, Amer. Math. Soc. Surveys **7**, (1961).
- [6] P. Dehornoy, *Groups with a complemented presentation*, J. Pure Appl. Algebra **116** (1997) 115–137.
- [7] P. Dehornoy, *Braids and Self-Distributivity*, Progress in Math. vol. 192, Birkhäuser, (2000).
- [8] P. Dehornoy, *Groupes de Garside*, Ann. Scient. Ec. Norm. Sup. **35** (2002) 267–306.
- [9] P. Dehornoy, *Complete positive group presentations*, J. of Algebra **268** (2003) 156–197.
- [10] P. Dehornoy & L. Paris, *Gaussian groups and Garside groups, two generalizations of Artin groups*, Proc. London Math. Soc. **79-3** (1999) 569–604.
- [11] P. Dehornoy, *Efficient solutions to the braid isotopy problem*, Disc. Appl. Math. **156** (2008) 3094–3112.
- [12] P. Dehornoy, *The subword reversing method*, Intern. J. Alg. and Comput. **21** (2011) 71–118.
- [13] P. Dehornoy, *Monoids of O-type, subword reversing, and ordered groups*, arXiv:1204.3211
- [14] P. Dehornoy, F. Digne, & J. Michel, *Garside families and Garside germs*, J. of Algebra, to appear, arXiv:1208.3362
- [15] P. Dehornoy, with F. Digne, E. Godelle, J. Michel, *Garside calculus*, in progress; <http://www.math.unicaen.fr/~garside/Garside.pdf>.
- [16] P. Deligne, *Les immeubles des groupes de tresses généralisés*, Invent. Math. **17** (1972) 273–302.
- [17] F. Digne & J. Michel, *Garside and locally Garside categories*, arXiv: 0612.652.
- [18] E. A. Elrifai & H.R. Morton, *Algorithms for positive braids*, Quart. J. Math. Oxford **45-2** (1994) 479–497.
- [19] D. Epstein, J. Cannon, D. Holt, S. Levy, M. Paterson & W. Thurston, *Word Processing in Groups*, Jones & Bartlett Publ. (1992).
- [20] F. A. Garside, *The braid group and other groups*, Quart. J. Math. Oxford **20-78** (1969) 235–254.
- [21] V. Gebhardt, *A new approach to the conjugacy problem in Garside groups*, J. Algebra **21** (2005) 282–302.
- [22] V. Gebhardt & J. González-Meneses, *Solving the conjugacy problem in Garside groups by cyclic sliding*, J. Symb. Comp. **28** (2010) 629–656.
- [23] A. Levy, *Basic Set Theory*, Springer Verlag (1979).
- [24] W. Thurston, *Finite state algorithms for the braid group*, Circulated notes (1988).

LABORATOIRE DE MATHÉMATIQUES NICOLAS ORESME, CNRS UMR 6139, UNIVERSITÉ DE CAEN, 14032 CAEN, FRANCE

E-mail address: patrick.dehornoy@unicaen.fr

URL: www.math.unicaen.fr/~dehornoy

SCHOOL OF COMPUTING, ENGINEERING, AND MATHEMATICS, UNIVERSITY OF WESTERN SYDNEY, LOCKED BAG 1797, PENRITH, NSW 2751, AUSTRALIA

E-mail address: v.gebhardt@uws.edu.au