Braids via term rewriting

Jörg Endrullis [a,*], Jan Willem Klop [a,b]

[a] *Vrije Universiteit Amsterdam, Department of Computer Science, Amsterdam, the Netherlands*
[b] *Centrum Wiskunde & Informatica (CWI), Amsterdam, the Netherlands*

A B S T R A C T

We present a brief introduction to braids, in particular simple positive braids, with a double emphasis: first, we focus on term rewriting techniques, in particular, reduction diagrams and decreasing diagrams. The second focus is our employment of the colored braid notation next to the more familiar Artin notation. Whereas the latter is a relative, position dependent, notation, the former is an absolute notation that seems more suitable for term rewriting techniques such as symbol tracing. Artin's equations translate in this notation to simple word inversions. With these points of departure we treat several basic properties of positive braids, in particular related to the word problem, confluence property, projection equivalence, and the congruence property. In our introduction the beautiful diamond known as the permutohedron plays a decisive role.

© 2018 Published by Elsevier B.V.

Braid theory is an area residing mostly in pure mathematics, where it occurs in a variety of areas, topology, group theory, homology theory, category theory, and also in mathematical physics and emerging theories in computer science such as quantum information. (See Abramsky [1] for many interconnections between such theories.) It is close to the border of mathematics and theoretical computer science, in particular to universal algebra and term rewriting.

Braids are interesting for the community of 'term rewriters' as they present several issues that are prominent in term rewriting theory: termination methods, word problems, completion methods, confluence proof methods, reduction diagram construction, and residual theory.

Vice versa, term rewriting is apparently also relevant for braid theory. This is in particular the case for reduction diagrams and for residual theory, originated in the study of lambda calculus, Combinatory Logic, and orthogonal rewrite systems. The main source for the origins of residual theory is the work of Huet and Lévy [2–5].

Lévy introduced for example the *Cube equation* that was fruitfully applied by Dehornoy [6,7], who showed how to combine these notions of residual theory with the familiar rewriting technique of reduction diagrams, constructed by tiling with elementary reduction diagrams, introduced in Klop [8,9] for term rewriting systems and lambda calculus; see also [10].

In this paper we will refer to this method to prove confluence as *confluence by tiling* [8,9]. This method now is known in braid theory as *word reversal*. A compendium of all these notions and techniques, pertaining to rewriting in general, can be found in *Terese* [10]. See also *Axiomatic Rewriting Theory VI* [11] by Paul-André Melliès.

One of the first occurrences of the notion of braids is in the notebooks of Gauss, as discovered in [12]. The theory of braids was put on the mathematical map by Emil Artin [13–15]. An important step was the work of Frank Garside, in his

---

ground-breaking thesis [16] and subsequent article [17]. The work of Artin and Garside was continued by the seminal work of Brieskorn and Saito [18].

For an extensive survey of classical and recent braid theory see Birman and Brendle [19]. Another recent much encompassing introduction to the deeper mathematics of braid theory is the book Kassel–Turaev [20]. However, in this chapter we will stay much more on the surface than those works. A short enjoyable popular description of braids and braid groups is contained in one of the books of Martin Gardner [21], where also some nice anecdotes are mentioned illustrating the significance of braids in quantum theory. An encompassing coverage of recent developments in braid theory and Garside theory is in the books by Dehornoy [6,7] and coworkers. In this paper we will occasionally refer to some similar notions and methods as treated in these works. Our paper will not present new results in braid theory; it is meant to be methodological, highlighting the connections mentioned above, with approaches from different sources that in spite of independent developments have led to very compatible notations and terminology.

1. Braid notations

The original topological description of braids is as follows:
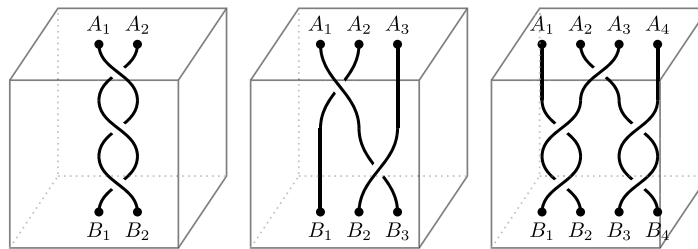
Each cube represents one braid, so there are 3 braids. For each braid, there is a finite sequence of initial positions, the dots numbered $1, \ldots, n$ in the upper row in the cube, and an equally long sequence of final positions $1, \ldots, n$ in a parallel row at the bottom of the cube. There are flexible strings (or strands) attached from the upper dots downwards to the final dots. The strings can be continuously deformed, but with the restriction that they may not leave the restricted space of the cube, and moreover, they may only 'go' downward and not bending upward again. They also should not intersect each other.

1.1. Artin's notation

If we 'flatten' the cubes above to a thinner slate of space in the viewing direction, it is clear that there is a way for the strings to cross each other 'over' or 'under', or depending on our viewing direction, 'before' or 'behind'. This is just as in the usual representation of knots as two-dimensional figures. Just as for knots, to suggest that a string is crossing behind (under) another string, is pictorially suggested by omitting a little bit of the string, as if it were invisible there.

We can stylise the drawing of braids even more by drawing them on a kind of music notation paper, with horizontal lines, as in:
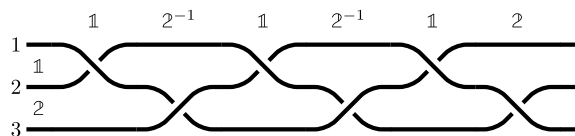

Now the Artin notation of braids assigns numbers $\mathbb{1}, \ldots, \mathbb{n} - \mathbb{1}$ to the *gaps* between consecutive strands, and then records with $\Bbbk$ a crossing in the $\Bbbk$-th gap that is 'positive' (the higher strand over the lower), $\Bbbk^{-1}$ the reverse, the higher strand under the lower. These crossings are then concatenated to form a *braid word*. For instance, the above braid is $\mathbb{1}\mathbb{2}^{-1}\mathbb{1}\mathbb{2}^{-1}\mathbb{1}\mathbb{2}$ in Artin's notation.

In this chapter we will only consider positive braid words, so all crossings are positive.

Definition 1.1 (*Positive braids*). A braid is *positive* if all crossings in Artin's notation are positive.

So when drawing the braid horizontally, for every crossing the upper strand crosses over the lower strand.

Example 1.2. For instance, the following braid is positive:
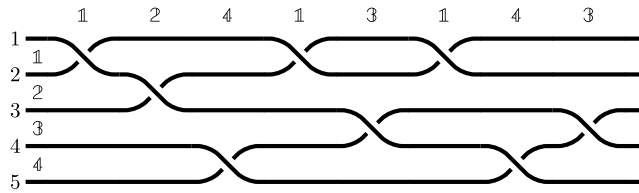
This braid corresponds to the braid word 12413143 in Artin's notation.

1.2. The colored braid notation: braid codes

We will now introduce an alternative notation for braids. It is mentioned in [22], with the name *colored braids presentation*. Consider the following positive 3-strand braid:
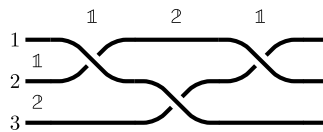
It is 121 in Artin's notation. We note that this notation is a *relative* notation with respect to which pair of strands is crossing: the first 1 in 121 signifies the crossing of strand 1 over 2, numbered from above, but the second 1 signifies the crossing of strand 2 over 3.

In the *colored braids presentation*, this relativity as to the occurrence of the symbols does not occur.

Definition 1.3 *(Braid codes).* In the colored braids presentation, a braid over n strands is represented by a word over alphabet

$$\{\alpha_{ij} \mid 1 \leq i, j \leq n, \ i \neq j\},$$

and a symbol α_{ij} stands for the i-th strand crossing over the j-th strand.

The braid 121 is then rendered as $\alpha_{12}\alpha_{13}\alpha_{23}$. So this braid code signifies as the first 'action' the crossing of strand 1 over 2, followed by crossing strand 1 over 3, and concluded with crossing 2 over 3.

To avoid repeated use of the phrase 'colored braid notation', we will refer to the words formed from the α_{ij}-symbols as *braid codes*, also for a clear distinction from the braid words in the usual rendering with symbols $1, 2, 3, \ldots$. It should be noted that this option is unusual in the standard mathematical approach, because it does not generalise to the case of some important classes of Artin monoids. From our term-rewriting inclined view it is favourable, because in term rewriting techniques, tracing symbols plays a prominent role.

In the sequel of this chapter we will make extensive use of the braid codes. Of course the notation is not as compact as Artin notation, but the crossing symbols α_{ij} facilitate the analysis of tracing symbols and drawing conclusions from such tracings, much better than the Artin notation. But the colored braid notation has also some disadvantages.

Remark 1.4. The following *caveat* should be noted for the colored braid notation. Consider positive 4-strand braids. In the relative notation, any word over the alphabet $\{1, 2, 3\}$ is a braid. For the colored braid notation, only a proper subset of the words over the symbols α_{ij} denotes a braid. For instance, α_{13} cannot occur at the beginning of the braid word. After an α_{23} we can have an α_{13}. Also, an α_{21} or an α_{32} cannot be the beginning. Inspecting the permutohedron, to be introduced later on, which displays both notations, confirms this. In Section 4 we will be more precise about which positive braid codes are 'well-formed'.

Remark 1.5. There is another representation for braids that employs generators α_{ij}, known as the BKL presentation; see [19, 22]. There α_{ij} indicates a general swap of strands i and j, not necessarily adjacent.

Remark 1.6. Both Artin's and colored braid notation are inspired by the well-known representations of the group S_n of permutations of $1, \ldots, n$.
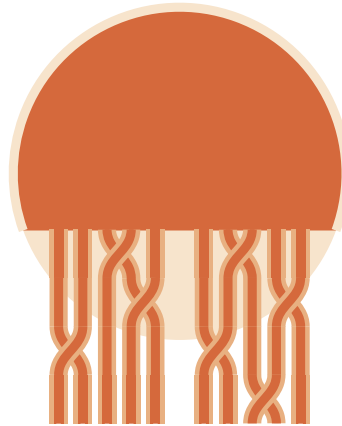
Fig. 1. Girl with braided hair: how to prolong the braids to the same result?

Let us now turn to some interesting problems for positive braids, namely the question when two braids are *equivalent* and the question whether positive braids are *confluent*.

The latter problem has been nicely described in Schmidt and Strohlein [23], in the following anthropomorphic words: A girl has two braids consisting of, say, 4 strings as shown in Fig. 1: The father starts braiding the left braid, the mother of the girl starts braiding the right braid. After some initial 'twists' they notice that they do it in a different way. But they want to arrive, eventually, at two identical braids. *Question:* can they go on and still arrive at braids that are the same? This is the question of *confluence*, we consider this question in Section 7. Before we can answer this question, we need to know:

 (i) What does it mean to continue braiding? Formally, this is a multiplication of braids; we consider braid *multiplication* in Section 2.
(ii) What does it mean that two braids that are the same? The question of braid *equivalence* has been decisively answered by Artin, see Section 3.

2. Braid multiplication

Braids can be concatenated or *multiplied*, denoted $a \cdot b$, as follows:
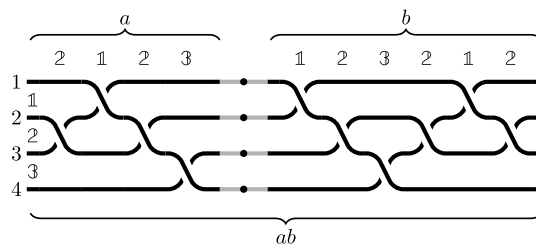
In the graphical representation, the product $a \cdot b$ is just the concatenation ab of the graphical representations of a and b. So multiplication coincides with word concatenation for Artin's notation. However, this is not the case for colored braid notation as we will see below. To avoid confusion we write concatenation of braid words without infix operator, so '·' is reserved for braid multiplication.

Remark 2.1. Actually, we could be more precise at this point. Whereas braid words in the usual rendering correspond to a braid *monoid*, this is no longer the case for braid codes in this paper with the α_{ij}-symbols. They ask for a more refined setting, and correspond to a *category* with as objects permutations of $\{1, \ldots, n\}$; two braid codes can be composed only if they correspond to matching permutations. In fact, this is the adopted categorical framework in Dehornoy [7].

2.1. Product in Artin's notation

In Artin's notation, the braid word ab is simply the concatenation of the braid words for a and b; we have

$$a = 2123$$

$$b = 123212$$

$$a \cdot b = 2123123212$$

2.2. Product in colored braid notation

For the colored braid notation, multiplication is more difficult. Due to the tracing of strands we need to take the permutation effect of a on b into account when computing $a \cdot b$. In the above example, let us numerate strands with $1, 2, 3, 4$ from top to bottom. In the colored braid notation we have:

$$a = \alpha_{23}\alpha_{13}\alpha_{12}\alpha_{14}$$

$$b = \alpha_{12}\alpha_{13}\alpha_{14}\alpha_{34}\alpha_{42}\alpha_{23}$$

Note that a causes the following permutation σ of the strands:

$$\sigma(1) = 4 \qquad \sigma(2) = 2 \qquad \sigma(3) = 1 \qquad \sigma(4) = 3$$

For computing $a \cdot b$ we need to apply the *inverse* of the permutation of a to b:

$$\sigma^{-1}(b) = \alpha_{32}\alpha_{34}\alpha_{31}\alpha_{41}\alpha_{12}\alpha_{24}$$

Then $a \cdot b$ is the concatenation of a with $\sigma^{-1}(b)$:

$$a \cdot b = \alpha_{23}\alpha_{13}\alpha_{12}\alpha_{14}\alpha_{32}\alpha_{34}\alpha_{31}\alpha_{41}\alpha_{12}\alpha_{24}$$

## 3. Equivalence of braids: Artin's equations

Two braids are *equivalent* if they can be transformed into each other by means of a continuous deformation[1] of the strands such that the strands never leave the cube and never intersect and the start and end points are kept fixed throughout. For instance the following two braids are equivalent:
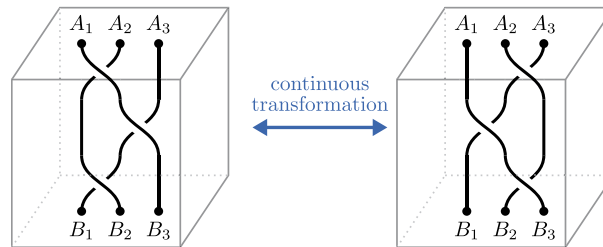
Think of the strands as rubber bands that are fixed on the top and the bottom. For the example of these two braids, it is not difficult to see that both braids can be transformed into each other by dragging the rubber bands inside the cube. While this topological definition of braid equivalence gives some intuition, it is not easy to work with.

3.1. Equivalence in Artin's notation

Artin has shown in his classical papers [13–15] that the topological equivalence can be characterised by simple equations on the braid representations. The braid 13 is 'the same', topologically viewed, as 31, just by shifting the crossings in the other order:
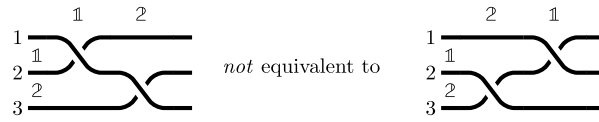

---
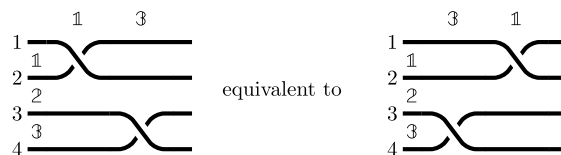
[1] See e.g. [24] for a precise description how a continuous deformation is 'modelled' by small discrete transformations.
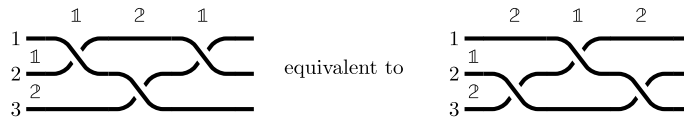
Also $1\!4$ is equivalent with $4\!1$. We will write $1\!3 = 3\!1$, and $1\!4 = 4\!1$. In general we have:

$$\hat{\imath}\,\hat{\jmath} = \hat{\jmath}\,\hat{\imath} \qquad \text{if } |\hat{\imath} - \hat{\jmath}| > 1$$

For adjacent gaps like 1 and 2, respective crossings do not commute:

But it is not hard to see that starting with $1\!2$ and $2\!1$, we can make them (topologically) equal by continuing $1\!2$ with 1 and $2\!1$ with 2. So $1\!2\!1 = 2\!1\!2$:

Note that $1\!2\!1$ and $2\!1\!2$ are indeed topologically the same; an experiment with actual strings of wire will demonstrate this. In fact, one of the so-called *Reidemeister moves* for the equivalence of knots is at stake here. In general we have:

$$\hat{\imath}\,\hat{\jmath}\,\hat{\imath} = \hat{\jmath}\,\hat{\imath}\,\hat{\jmath} \qquad \text{if } |\hat{\imath} - \hat{\jmath}| = 1$$

The equations above completely define the topological equivalence considered (see Artin [13–15]). Historically, these relations were proved by Artin to be a representation of the braid *group*; that they also constitute a presentation of the positive braid *monoid* was proved only in 1967 by Garside. For more background, see [25]. For an authoritative introduction to the algebraic theory of positive braids, together with complexity considerations of interest to computer scientists, see [26, Chapter 9].

Theorem 3.1 *(Positive braid relations). Two positive braids with n strands are (topologically) equivalent if and only if the corresponding braid words in Artin's notation are equal modulo the following system of equations:*

$$\hat{\imath}\,\hat{\jmath} = \hat{\jmath}\,\hat{\imath} \qquad\qquad \textit{if } |\hat{\imath} - \hat{\jmath}| > 1 \tag{1}$$

$$\hat{\imath}\,\hat{\jmath}\,\hat{\imath} = \hat{\jmath}\,\hat{\imath}\,\hat{\jmath} \qquad\qquad \textit{if } |\hat{\imath} - \hat{\jmath}| = 1 \tag{2}$$

for all $\hat{\imath}, \hat{\jmath} \in \{1, \ldots, n\}$.

Example 3.2. In the following example in Fig. 2 we give the convertibility class of $\Delta_4 = 1\!2\!3\!1\!2\!1$. We also give there the colored braid notation that will be introduced next:

3.2. Equivalence in colored braid notation

Using braid codes, the relations of Artin and, for positive braids, Garside, take a different pleasant form. For instance $1\!2\!1 = 2\!1\!2$ becomes

$$\alpha_{12}\alpha_{13}\alpha_{23} = \alpha_{23}\alpha_{13}\alpha_{12}$$
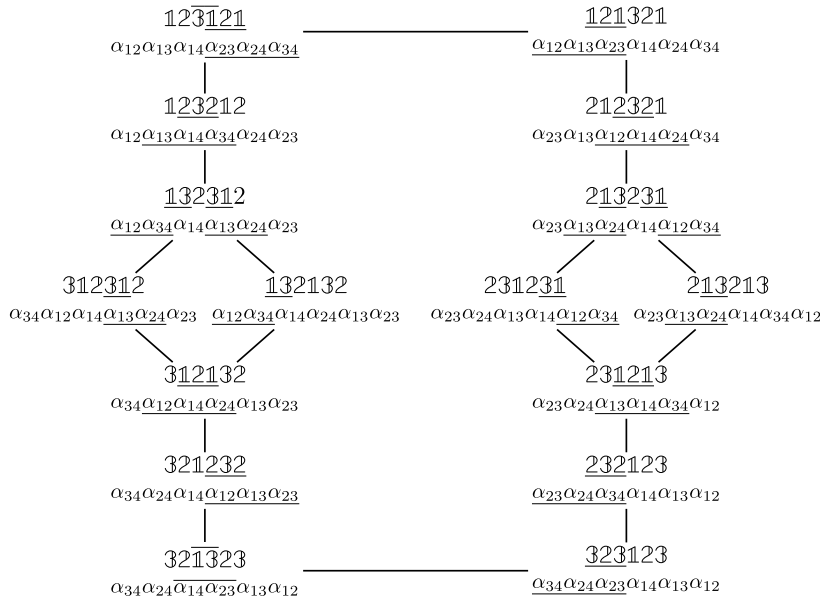
For braids with n strands, the braid equations are now (see Bangert [22]):

Theorem 3.3 *(Braid equivalence for braid codes). Two positive braid diagrams with n strands are (topologically) equivalent if and only if the corresponding braid codes are equal modulo the following system of equations between braid codes:*

$$\alpha_{kl}\alpha_{ij} = \alpha_{ij}\alpha_{kl} \qquad\qquad \textit{if } i, j > k, l \tag{3}$$

$$\alpha_{ij}\alpha_{ik}\alpha_{jk} = \alpha_{jk}\alpha_{ik}\alpha_{ij} \tag{4}$$

for all pairwise distinct $i, j, k, l \in \{1, \ldots, n\}$.

$$
\begin{array}{cc}
123121 & 121321 \\
\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{23}\alpha_{24}\alpha_{34} & \alpha_{12}\alpha_{13}\alpha_{23}\alpha_{14}\alpha_{24}\alpha_{34} \\[4pt]
123212 & 212321 \\
\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{34}\alpha_{24}\alpha_{23} & \alpha_{23}\alpha_{13}\alpha_{12}\alpha_{14}\alpha_{24}\alpha_{34} \\[4pt]
132312 & 213231 \\
\alpha_{12}\alpha_{34}\alpha_{14}\alpha_{13}\alpha_{24}\alpha_{23} & \alpha_{23}\alpha_{13}\alpha_{24}\alpha_{14}\alpha_{12}\alpha_{34}
\end{array}
$$

$$
\begin{array}{cccc}
312312 & 132132 & 231231 & 213213 \\
\alpha_{34}\alpha_{12}\alpha_{14}\alpha_{13}\alpha_{24}\alpha_{23} & \alpha_{12}\alpha_{34}\alpha_{14}\alpha_{24}\alpha_{13}\alpha_{23} & \alpha_{23}\alpha_{24}\alpha_{13}\alpha_{14}\alpha_{12}\alpha_{34} & \alpha_{23}\alpha_{13}\alpha_{24}\alpha_{14}\alpha_{34}\alpha_{12}
\end{array}
$$

$$
\begin{array}{cc}
312132 & 231213 \\
\alpha_{34}\alpha_{12}\alpha_{14}\alpha_{24}\alpha_{13}\alpha_{23} & \alpha_{23}\alpha_{24}\alpha_{13}\alpha_{14}\alpha_{34}\alpha_{12} \\[4pt]
321232 & 232123 \\
\alpha_{34}\alpha_{24}\alpha_{14}\alpha_{12}\alpha_{13}\alpha_{23} & \alpha_{23}\alpha_{24}\alpha_{34}\alpha_{14}\alpha_{13}\alpha_{12} \\[4pt]
321323 & 323123 \\
\alpha_{34}\alpha_{24}\alpha_{14}\alpha_{23}\alpha_{13}\alpha_{12} & \alpha_{34}\alpha_{24}\alpha_{23}\alpha_{14}\alpha_{13}\alpha_{12}
\end{array}
$$

Fig. 2. The connections denote conversion steps between the 16 positive simple braid words convertible with Garside's fundamental word Δ_4. There are two symmetries: left–right is inversion for Artin's notation and up–down is Garside's renaming R, swapping the generators $\mathbb{1},\mathbb{2},\mathbb{3}$ to $\mathbb{3},\mathbb{2},\mathbb{1}$, so $R(\mathbb{1}) = \mathbb{3}$, $R(\mathbb{2}) = \mathbb{2}$, $R(\mathbb{3}) = \mathbb{1}$. For the braid codes rendering, the symmetries are a bit more complicated.

We write $x \approx y$ if x is convertible with y by equational reasoning using Artin's equations (both for Artin's notation and for braid codes).

The equation (4) occurs in several areas, e.g. quantum groups, and is often called the *Yang–Baxter equation*. The equations in the relative notation, $\mathbb{121} = \mathbb{212}$, etc., are sometimes also called by that name.

So the braid axioms are now just inversions of certain factors in braid codes.

Definition 3.4 (*Inversion*). Let $w = a_1 a_2 \cdots a_n$ be a word in Σ^* for some alphabet Σ, then $\mathrm{inv}(w) = a_n a_{n-1} \cdots a_1$, the *inversion* of w.

Proposition 3.5. *Both in Artin and braid code notation:*

$$\alpha \approx \beta \quad \Leftrightarrow \quad \mathrm{inv}(\alpha) \approx \mathrm{inv}(\beta)$$

Proof. Obvious since the axioms are invariant under inversion. □

An extensive example is given in Fig. 2 which will also be relevant when we study the permutohedron P_4 later on.

Remark 3.6. Fig. 2 illustrates the following three observations that have easy proofs, left to the reader:

  (i) Conversion and translation (between Artin and braid code notation) commute.
 (ii) Inversion and conversion commute, both for Artin and braid code notation.
(iii) However, inversion and translation notation do not commute.

Before continuing our development of the basic theory both for the usual Artin notation of braid words, and for our alternative version of braid codes, it is time for a brief recapitulation of the nontrivial ontology of notions in the set-up so far. There are several types of objects involved. In Fig. 3 we have surveyed these types of objects in an entity-relationship diagram.

Note how striking it is that purely continuous topological notions can be captured fully in the discrete setting of words, word rewriting, finite automata, complete string rewrite systems, and corresponding algorithms to efficiently compute normal forms.

4. Translating between Artin notation and braid codes

Using string diagrams for positive braid words as in the preceding figures, it is easy to convert Artin notation involving $\mathbb{1}, \mathbb{2}, \mathbb{3}, \ldots$ into braid codes involving $\alpha_{12}, \alpha_{23}, \ldots$. But in general this is too cumbersome and we need a simple algorithm
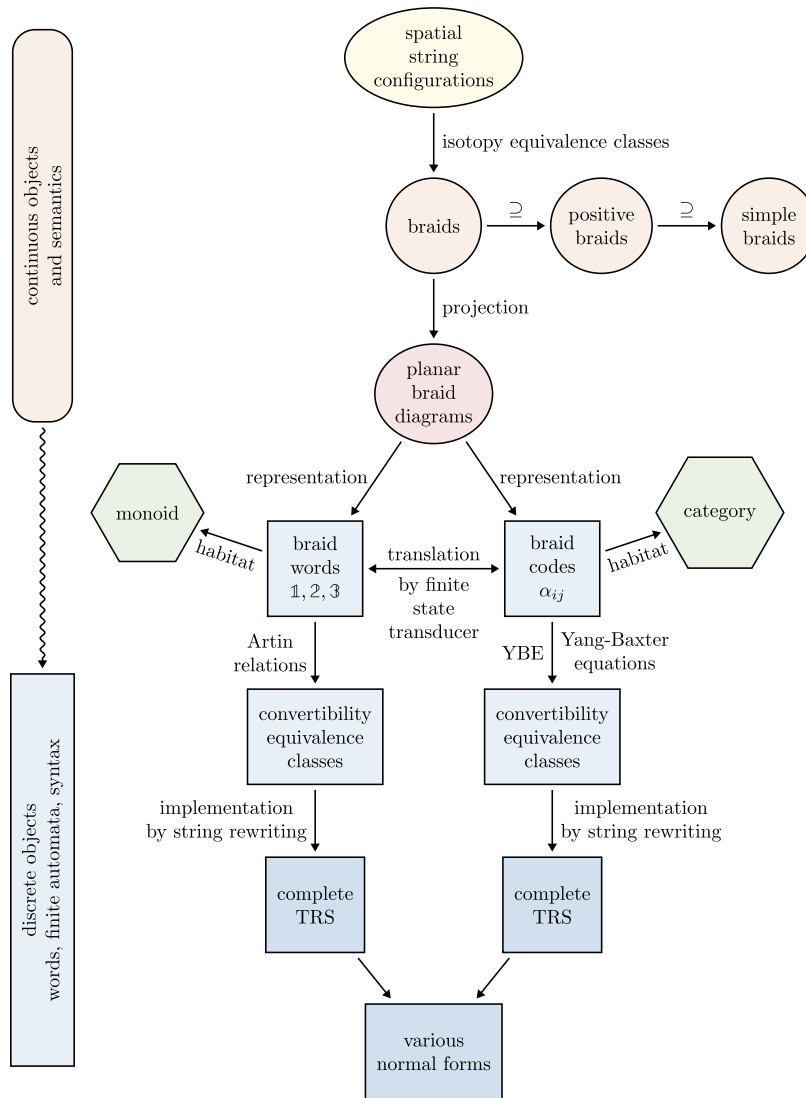
Fig. 3. Ontology of braids: from continuous to discrete.

for interchanging notations, and also to see which braid codes are well-formed. In fact these braid codes constitute a regular language. We will now give a *finite state automaton* (FSA) for this regular language and a *finite state transducer* (FST) [27–29] for interchanging Artin notation and braid codes. In fact, this FSA and FST present themselves in a very easy way; they are known as the permutohedron of order n, P_n for short. The permutohedron P_2 is a line segment, P_3 is the hexagon in Fig. 4 and P_4 is in Fig. 6.

5. Simple braid words

There is a particular important subset of braids, consisting of the *simple* braids, that is braids that not have multiple crossings of the same pair of strands. With the braid codes we can state this more precisely.

Definition 5.1 (*Simple braid codes*). A braid code $\alpha_{i_1 j_1} \alpha_{i_2 j_2} \cdots \alpha_{i_m j_m}$ is *simple* if for every $1 \le k < l \le m$ we have $\{i_k, j_k\} \neq \{i_l, j_l\}$.

So a simple braid code contains neither: two occurrences of α_{ij}, nor both α_{ij} and α_{ji}. For instance, the braid code $\alpha_{12} \alpha_{13} \alpha_{31}$ is not simple.

As we are only interested in positive braids, we moreover require simple braids to be positive.
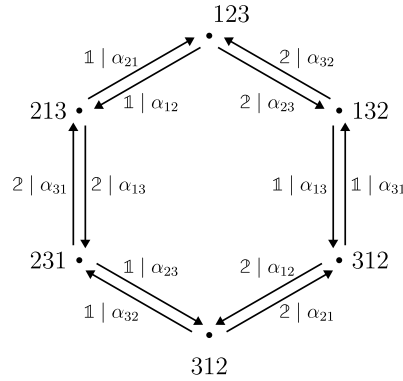
Fig. 4. Permutohedron P_3 as the finite state transducer translating Artin's notation of braids with 3 strands into braid codes. A braid word in Artin's notation is entered at the top 123 and translated following the arrows, registering at each step the translation instruction $a \mid b$. Also the reverse translation, after flipping each $a \mid b$ into $b \mid a$.
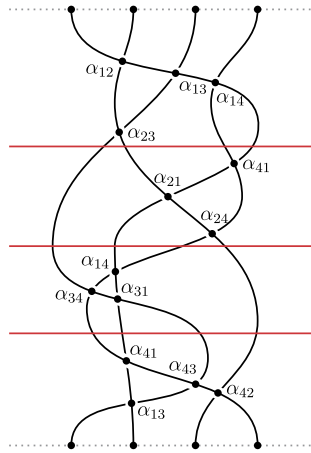
Fig. 5. Positive non-simple braid partitioned in four simple braids.

Definition 5.2 *(Simple braid).* A braid is *simple* if it is positive and corresponds to a simple braid code.

The salient feature of simple braids is that their convertibility classes correspond 1–1 with permutations of $1, 2, 3, \ldots, n$, for braids working on n strands.

Theorem 5.3. *Every positive braid is the product of simple braids. Let A be a positive braid, then there exist $n \in \mathbb{N}$ and simple braids $B_1, B_2, \ldots, B_n$ such that*

$$A = B_1 \cdot B_2 \cdot \cdots \cdot B_n$$

Example 5.4. Fig. 5 gives an example how to cut up a braid into simple braid code constituents:

$$\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{23} \quad \alpha_{41}\alpha_{21}\alpha_{24} \quad \alpha_{14}\alpha_{34}\alpha_{31} \quad \alpha_{41}\alpha_{43}\alpha_{42}\alpha_{13}$$

$$= \alpha_{12}\alpha_{13}\alpha_{14}\alpha_{23} \cdot \alpha_{34}\alpha_{24}\alpha_{23} \cdot \alpha_{23}\alpha_{13}\alpha_{12} \cdot \alpha_{12}\alpha_{13}\alpha_{14}\alpha_{23}$$

Here we have indicated the cutting points by extra spacing.

Remark 5.5. Cutting up a positive braid word in factors that are simple, is very much akin to the well-known method in λ-calculus and term rewriting of cutting up a whole reduction (rewrite) sequence into simple pieces, known as *developments*; see [10, Chapter 4] and [30]. In such a development, no created redex may be contracted; in a simple braid word, no earlier crossing may be repeated.
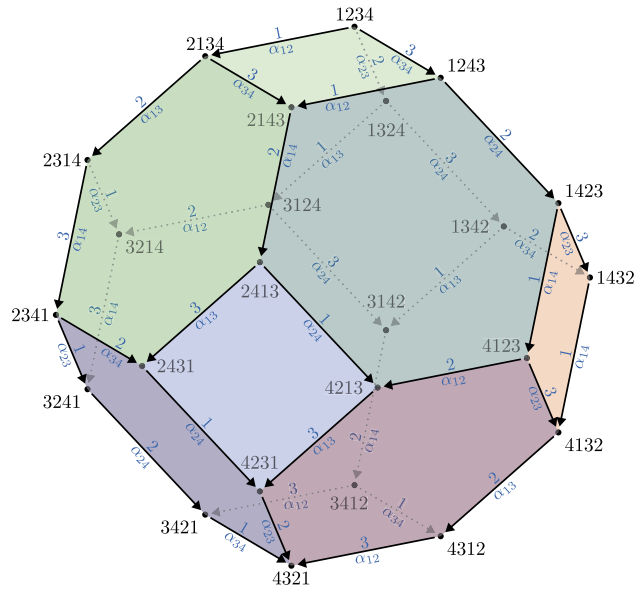
Fig. 6. Permutohedron of order 4 with Artin notation and colored braid notation. The 24 vertices contain the permutations of $1, 2, 3, 4$, the elements of the symmetric group S_4. The edges are labelled with the elementary transpositions generating S_4. The 14 facets, 6 squares and 8 hexagons, are identical to the non-trivial elementary braid diagrams in Definition 7.2.

Remark 5.6. We note that the minimal number of cuts into simple parts depends on which word in the conversion class is considered, as also noted in [7, Example 1.11]. For instance:

$$11335577 = 1 \cdot 13 \cdot 35 \cdot 57 \cdot 7$$

$$\approx 13571357 = 1357 \cdot 1357$$

6. The permutohedron

We will now investigate the *permutohedron*, which is a thing of beauty and also a key to much of the basics for braids.

In Garside [16,17] it is mentioned that the Cayley graph of the fundamental word Δ_4 is the "2-skeleton" of the truncated dodecahedron, which is also known as the permutohedron.

Definition 6.1. For a braid with $N + 1$ strands, so N gaps, the Garside element Δ_N is defined as follows:

(i) $\Pi_s \equiv 12 \cdots s$.
(ii) $\Delta_N \equiv \Pi_N \Pi_{N-1} \cdots \Pi_2 \Pi_1$.

In braid code notation, Garside's fundamental word Δ_N is represented by

$$\prod_{i=1}^{n-1} \prod_{j=i+1}^{n} \alpha_{ij}$$

So,

(i) Δ_2 is represented by α_{12},
(ii) Δ_3 by $\alpha_{12}\alpha_{13}\alpha_{23}$ (see also Fig. 4), and
(iii) Δ_4 by $\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{23}\alpha_{24}\alpha_{34}$ (see also Figs. 2 and 6).

The permutohedron can be rendered as a 3-dimensional polytope as displayed in Fig. 6. We have enriched the 24 nodes of the permutohedron with the images of the corresponding permutations of the original sequence 1234, which decorates the top of this structure, which is known to be a complete lattice. The bottom is the swapped sequence 4321. On the edges there are the generators $\mathbb{1}, \mathbb{2}, \mathbb{3}$ of Artin's notation of braids, and also the corresponding α_{ij} for the braid codes. There are 6 squares and 8 hexagons and these 14 facets demarcate the 16 simple braid words in both notations, mutually convertible, representing the Garside element Δ_4, displayed in Fig. 2, in Artin's notation and as braid codes.

We note in advance that the squares and hexagons are precisely the two elementary diagrams that will be encountered in Section 7 about diagram constructions.

Remark 6.2.

(i) Note the symmetry with respect to the centre of the sphere: not only the 'states' $1234, \ldots, 4321$ are mirrored, also the 'transitions' $\alpha_{12}, \alpha_{23}, \ldots$ (6 in number) are preserved in this symmetry, both in Artin's and in colored notation.

(ii) Apart from these obvious symmetries, the permutohedron has the group of permutations S_4 as group of symmetries: Given two of its nodes s_1 and s_2, then the permutation transposing the label of s_1 into that of s_2, and performed on all vertices, constitutes a symmetry of P_4 mapping s_1 to s_2. In other words, P_4 is *vertex-transitive* with respect to the action of S_4 on P_4 as described.

(iii) A third symmetry of P_4 (and of P_n in general) is that it is 'edge-label transitive' in the following sense. Consider the node $s = 3241 \in P_4$ and use the corresponding permutation ρ as a rotation of P_4 such that s becomes the top 1234. Then the Artin labels $\mathbb{1}, \mathbb{2}, \mathbb{3}$ are invariant under ρ:

$$s_1 \overset{\mathbb{i}}{\to} s_2 \quad \Rightarrow \quad \rho(s_1) \overset{\mathbb{i}}{\to} \rho(s_2)$$

The braid code generators α_{ij} are renamed

$$s_1 \overset{\alpha_{ij}}{\to} s_2 \quad \Rightarrow \quad \rho(s_1) \overset{\alpha_{\rho(i)\rho(j)}}{\to} \rho(s_2) \,.$$

(iv) The order on P_4 from top 1234 to bottom 4321 is also called the *Bruhat order*.

A remarkable feature of P_4 is that it tessellates the 3-dimensional space $\mathbb{R}^3$. Analogous facts hold for P_{n+1}, tessellating $\mathbb{R}^n$, $n = 1, 2, \ldots$. Indeed, the unit segment P_2 tessellates the whole line $\mathbb{R}$, just as the hexagon P_3 tessellates the plane $\mathbb{R}^2$. That P_5 tessellates $\mathbb{R}^4$ is harder to see. We wonder if this tessellation property has a significance for braids.

The permutohedron possesses an interesting property, that we will call the 'homotopy property', because it is reminiscent of that notion in algebraic topology.

Proposition 6.3 (Homotopy property). *Any two paths on P_n having the same start point s_1 and end point s_2, and such that each step decreases the Bruhat order, are convertible.*

The well-known Bruhat order $<$ is defined on permutations by

$$\sigma < \tau \quad \Leftrightarrow \quad \mathrm{swaps}(\tau) \subsetneq \mathrm{swaps}(\sigma)$$

where $\mathrm{swaps}(\sigma) = \{ \langle i, j \rangle \mid i < j, \ \sigma(i) > \sigma(j) \}$. Equivalently, on simple braids we have $x > y$ if $\exists z.\, xz = y$.

The permutohedron is such a wonderful object that we like to contemplate it a bit more. As we have seen, it is the domain of simple positive braids. Enriched with the two styles of notations, the Artin notation and the braid codes, it is also a finite state transducer, translating the two notations into each other, not only for simple but for general positive braid words. The permutohedron is also suitable as a location for general positive braid words, not only simple ones that start from the north pole 1234 to the south pole 4321, all the way or part of the way.

There are 16 ways to go from 1234 to 4321 if we go only downwards. See Fig. 2. But we can also enter general positive braid words starting from 1234 when we use the back arrows (also labelled with $\mathbb{1}, \mathbb{2}, \mathbb{3}$), or as braid codes with α_{ji} for a reversed α_{ij} arrow. The general positive braid word can then travel as a 'curve' all over the globe P_4, possibly with cycles. Even so, we can recognise which factors of the braid word are simple: they are the parts of the curve where the arrows are uni-directional. Only when the direction of the travel of the curve is altered, a simple part is ending, and a new simple part is starting. Thus e.g. the cyclic curve $\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\cdots$ splits in a new simple part after each step, all just one step $\mathbb{1}$.

Braid words thus travelling over the globe P_4 in whatever way, can be 'continuously' transformed into others, where a part of the word is nudged or swapped in a 4 or 6-cell to the other side.

Garside has, as is well-known, a beautiful theorem stating that any positive braid 'curve', travelling over P_n, can by nudging in the elementary cells (or elementary diagrams as we have used), be transformed to wind itself a certain number of times around the globe P_n, followed by some tail that cannot make a total orbit around the globe; the tail is then 'prime' to Δ. Primeness is easy to relate to the geometry of the globe P_n: if the curve does not travel through two antipodal points on the globe, it is prime to Δ.

Thus any positive braid has a unique 'winding number' n and a unique tail modulo convertibility. A braid thus is reminiscent of a wave: it contains a fixed number of maximal waves, followed by a tail of small waves. The medium that is oscillating, is the state space $1234, \ldots, 4321$. Moreover the big waves commute with the small waves, they can be at will preponed or postponed as Garside demonstrated in his algebraic calculations in [17]:

$$\mathbb{1}\Delta \approx \Delta\mathbb{3} \qquad \mathbb{2}\Delta \approx \Delta\mathbb{2} \qquad \mathbb{3}\Delta \approx \Delta\mathbb{1}$$

for P_4, and for general P_n:

$$\triangle\triangle\mathring{\mathrm{i}} \approx \mathring{\mathrm{i}}\triangle\triangle$$

The verification is omitted here, it is in Garside [17] and many introductions to braid theory. It is a simple exercise.

7. Confluence and equivalence via reduction diagrams

The confluence problem is now: given two elements u, v of this braid semi-group, can we always find elements x, y such that $ux \approx vy$?

Theorem 7.1 *(Confluence). Positive braids are confluent, that is*

$$\forall u, v . \exists x, y . ux \approx vy$$

This was first proven by Garside. Actually he proved $\forall u, v . \exists x, y . xu \approx yv$, but this is easily seen to be equivalent to confluence using Proposition 3.5. Garside's proof involved an ingenious computation for general positive braids, relying on the use of his fundamental word $\triangle_n$. We will follow a quite different road, employing reduction diagrams.

Reduction diagrams, a familiar technique from term rewriting, allow for an elegant proof of confluence of positive braids. Reduction diagrams have been used in many of the early papers on the lambda calculus, and in the more general theory of term rewriting systems. In Klop [8] *reduction diagrams with empty steps* were introduced. These reduction diagrams are built by gluing together simple *elementary diagrams*. For braids we use the elementary diagrams as shown in Definition 7.2. They are just a graphical way of rendering the defining equations for braid equivalence. We have also included the necessary trivial elementary diagrams that involve empty sides, including the one with all sides empty.

Definition 7.2 *(Elementary diagrams for braids).* For a braid with n strands, we have the following elementary diagrams
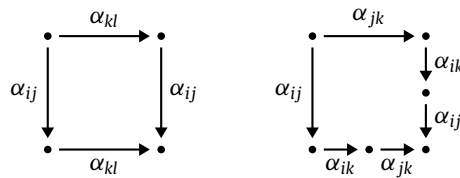
for pairwise distinct $i, j, k, l \in \{1, \ldots, n\}$, and trivial elementary diagrams
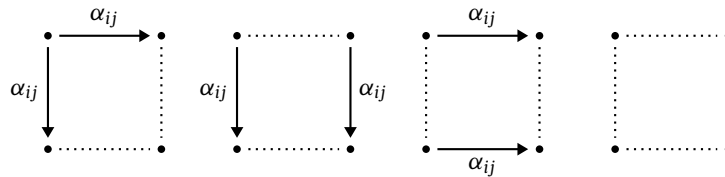
for $i, j \in \{1, \ldots, n\}$, $i \neq j$. Here the dotted lines without arrowhead stand for empty steps.

The (non-trivial) diagrams express the braid code equations $\alpha_{ij}\alpha_{kl} = \alpha_{kl}\alpha_{ij}$ and $\alpha_{ij}\alpha_{ik}\alpha_{jk} = \alpha_{jk}\alpha_{ik}\alpha_{ij}$. The trivial elementary diagrams stand for trivial equations such as $\varepsilon\alpha_{ij} = \alpha_{ij}\varepsilon$ and $\varepsilon\varepsilon = \varepsilon\varepsilon$, where ε is the empty step, functioning as a unit element.

Remark 7.3 *(Elementary diagrams in Artin's notation).* In Artin's notation the elementary diagrams for braids with n strands look as follows:
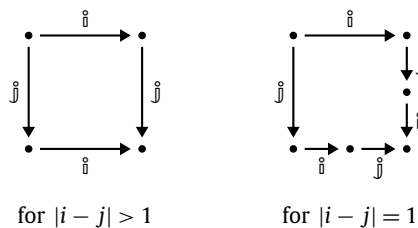
for $|i - j| > 1$ for $|i - j| = 1$

and trivial elementary diagrams
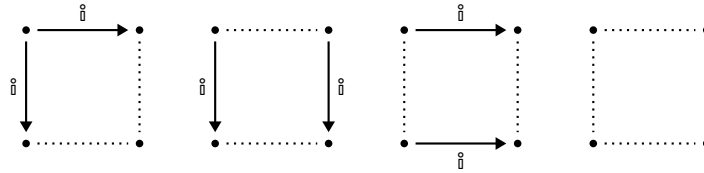
for $i, j \in \{1, \ldots, n\}$.

In the theory of term rewriting such elementary diagrams are familiar for confluence proofs. Confluence is obtained when tiling an initial pair of finite, divergent reduction sequences, leads to a completed reduction diagram with converging sides that are tantamount to confluence. So we now naturally apply the tiling effort in the present issue of confluence for braids.

The elementary diagrams are scalable, both horizontally and vertically, so they can be glued together with adjacent diagrams having multiple steps. Now confluence of braids is obtained by a simple *tiling game* as illustrated in the following example.

Example 7.4. Let us consider two braid codes with 4 strands:

$$\alpha_{23}\alpha_{24}\alpha_{13} \qquad \text{and} \qquad \alpha_{12}\alpha_{13}$$

We start with an empty reduction diagram where the horizontal reduction corresponds to one of the two braid words, and the vertical reduction to the other:
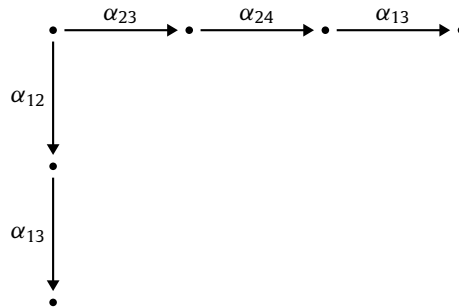
We now *tile* this reduction diagram by pasting matching elementary diagrams. We start with the upper-left corner:
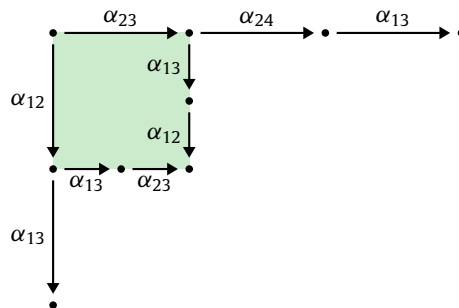
There are now two peaks ⌐→ and for both we paste the matching elementary diagrams:
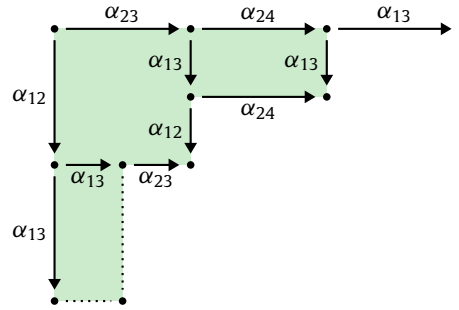
Here the dotted lines without an arrowhead indicate empty steps.

We continue pasting elementary diagrams until there are no more peaks left and the entire diagram is completed:
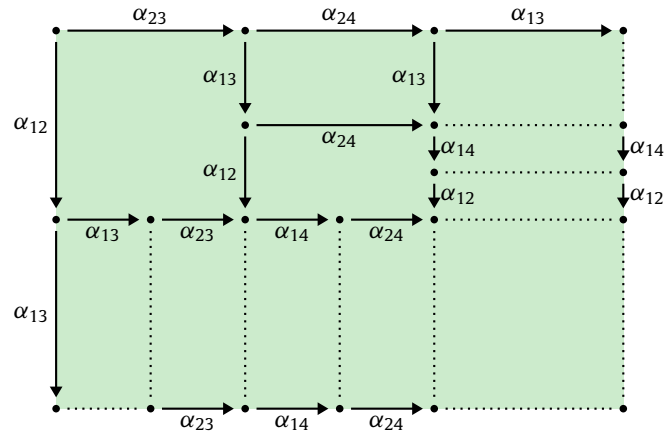
This process is somewhat reminiscent of covering a floor with tiles. Therefore it is frequently referred to as *tiling*.

Example 7.5. The tiling shown in Fig. 7 is a completed reduction diagram solving the parent's problem in Fig. 1. The bottom and the right side of the diagram yield the confluent braid word extensions.

Notation 7.6. For a relation $\to \ \subseteq A \times A$ we write

(i) $\twoheadrightarrow$ or $\to^*$ for the reflexive, transitive closure of $\to$, and
(ii) $\to^=$ for the reflexive closure of $\to$.

Definition 7.7 (*Confluence by tiling*). Let A be a set and $\to \ \subseteq A \times A$ be a relation. A *complete set of elementary diagrams* consists for every peak $b \leftarrow a \to c$ of elementary diagrams of the form
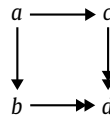
for some d, and it contains trivial elementary diagram of the form
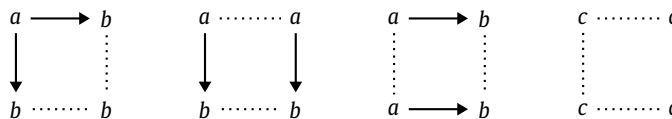
for every $a \to b$ and $c \in A$. Here the dotted lines stand for empty steps. *Tiling* is the process of repeated adjunction of elementary diagrams to a partially completed reduction diagram.
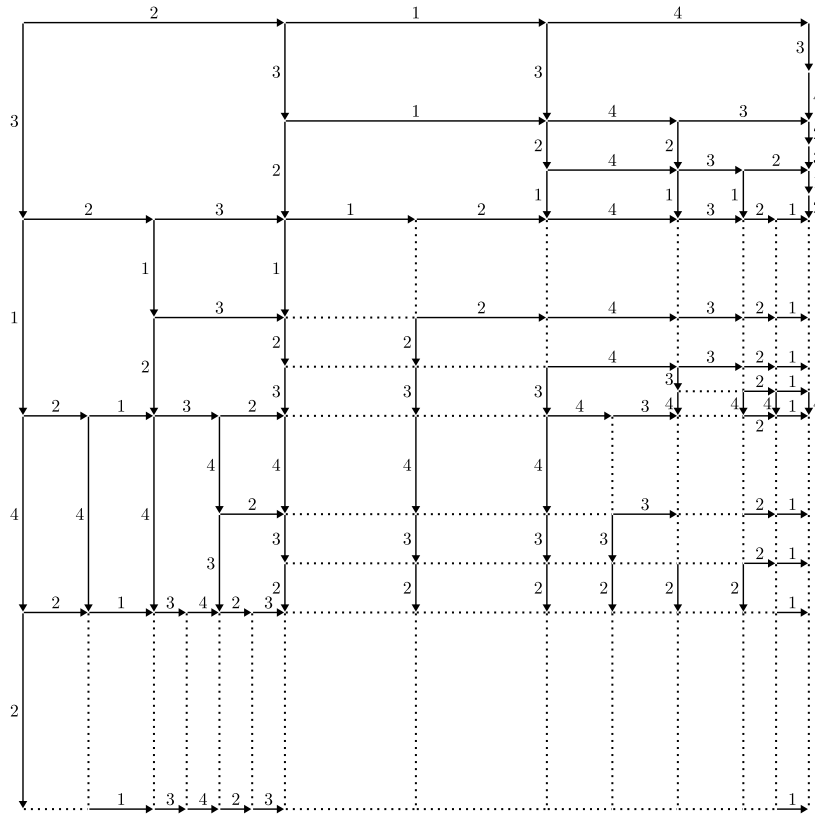
Fig. 7. Solution of the parent's braiding problem in Fig. 1.

If tiling is terminating using some complete set of elementary diagrams is terminating, then → is said to be *confluent by tiling*[2] (with respect to this set of elementary diagrams).

It will always be clear what set of elementary diagrams we are using, so we will omit explicit mention of this set and just speak of 'confluence by tiling'.

Proposition 7.8. *Confluence by tiling implies confluence.*

Proof. Obvious: the right and bottom side of the completed reduction diagram provide confluent joining reductions. □

The rationale of the trivial elementary diagrams with the empty steps is first to keep reduction diagrams in a rectangular orthogonal shape which facilitates tracing of symbols inside such a diagram. Another reason is that the first trivial elementary diagram which expresses absorption of identical steps is instrumental in comparing reduction sequences as to the 'work' done by crossing out common steps against each other; see later on the notion of Lévy equivalence. One might think that the empty steps could present a complication with respect to termination of tiling. But it is an easy exercise to prove that an infinite reduction diagram must possess an infinite proper reduction, that is one without empty steps. An appeal to König's Lemma will readily yield this fact.

7.1. Completeness and uniqueness

The set of elementary diagrams from Definition 7.2 is *complete* in the sense that we cannot get stuck during tiling. There always is a unique matching elementary diagram for every peak of compatible steps.

Definition 7.9. Two braid codes u and v are *compatible* if there exists a braid code w such that wu and wv are positive braid codes.

---

[2] In [8, Section 6.1, pages 58–69], confluence by tiling is called CR^+ and this property is established for lambda calculus and extensions. See further also [31].

Example 7.10. The steps α_{12} and α_{13} are not compatible, they cannot be an extension of the same braid code. Either strand 1 is directly left of strand 2 or strand 3, but not both at the same time.
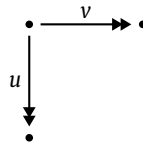
Lemma 7.11 (*Completeness and uniqueness of the elementary diagrams*). *For every pair of compatible braid codes $u \in \{\alpha_{ij}, \varepsilon\}$ and $v \in \{\alpha_{kl}, \varepsilon\}$ there exists precisely one elementary diagram of the form*
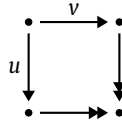
in Definition 7.2.

Proof. Straightforward case analysis. □

Assume that we are tiling a reduction diagram

for compatible braid codes u and v. Then all peaks ⌐→ encountered during tiling arise from compatible steps, and thus can be filled by a matching elementary diagram. This ensures that we can continue tiling until there are no more peaks, and the reduction diagram is completed.

But is the process of tiling guaranteed to terminate? Does it stop after a finite number of steps? Apparently, we were lucky in the above examples that the tiling procedure was terminating. But how do we prove that this is so in general? We give two proofs of termination in Sections 7.4 and 7.5.

Before we prove termination, we observe that completed reduction diagrams are unique.

Lemma 7.12 (*Determinism of tiling*). *Let u, v be compatible positive braid codes. If there exists a completed reduction diagram*
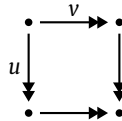

then every way of tiling
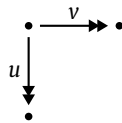
will terminate in the same completed reduction diagram.

Proof. Let D be the completed reduction diagram. The non-determinism in tiling arises from the choice which peak to fill next. For each single peak ⌐→ there is a unique elementary diagram by Lemma 7.11. So locally the choice is deterministic. This implies the following invariant during tiling: every partially completed reduction diagram is a subdiagram of D. Furthermore, the number of tiling steps is bounded by the number of tiles in D. Hence every tiling terminates with result D. □

7.2. Path equivalence

We start with an important observation stating that paths in a reduction diagram having the same start and end points are convertible with respect Artin's equations. The way to retrieve the conversion from the diagram employs a view used in Klop [8] for lambda calculus, elaborated axiomatically in Melliès' series of foundational papers for lambda calculus and rewriting [11]. This is the dual view of reading an elementary reduction diagram:
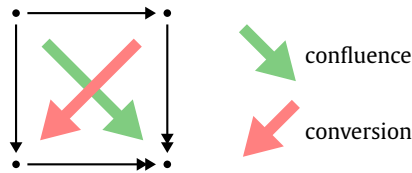
A *path* in a reduction diagram is an alternation of nodes and steps with the understanding that the steps follow the arrows or empty steps in the direction right or down. If we speak of a path u then u is the word obtained by concatenating the labels encountered on the steps (the empty word ε for empty steps).

Theorem 7.13 *(Path equivalence in reduction diagrams). In a (partially completed) reduction diagram, all paths having the same start and end point are equivalent modulo Artin's equations (or its braid code version).*[3]

Proof. We prove the theorem by induction on the number of elementary diagrams in the area surrounded by the paths. In this proof, we write an arrow $\to$ for steps in the reduction diagram, tacitly including empty steps. Sot let ρ_1 and ρ_1 be paths having the same start and end points in the reduction diagram. This can be illustrated as follows:
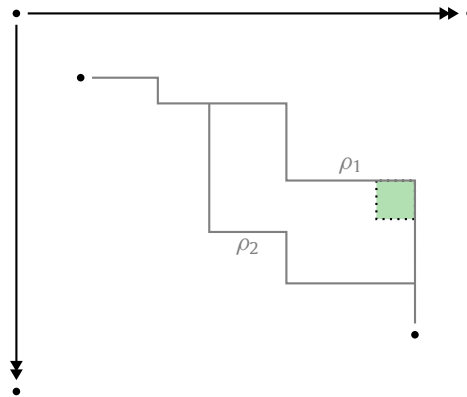

If the surrounded area is 0, then both paths are identical ($\rho_1 = \rho_2$).

So assume that the surrounded area is not empty. Then there exists a tile (an elementary diagram)
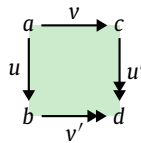
in the surrounded area such that $a \xrightarrow{v} c \xrightarrow{u} d$ is a part of ρ_1 or ρ_2; without loss of generality say ρ_1. In the above re- duction diagram this tile is highlighted (in green). Then we have $vu' \approx uv'$ as this holds for every elementary diagram (Definition 7.2). Let ρ_1' be obtained from ρ_1 by replacing
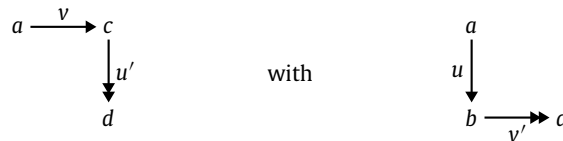

Then $\rho_1 \approx \rho_1'$ and the area between ρ_1' and ρ_2 is smaller than the area between ρ_1 and ρ_2. Thus by induction hypothesis we have $\rho_1' \approx \rho_2$ and hence $\rho_1 \approx \rho_2$.

Above we have given an intuitive, visual argument that there is a tile in the surrounded area with the claimed properties. We now give a more formal argument. Consider a node x where the paths ρ_1 and ρ_2 join; so there are steps $y \to x$ in ρ_1 and $z \to x$ in ρ_2 such that $y \neq z$. Then one of these steps is horizontal (to the right) and the other one is vertical (downwards). Without loss of generality we may assume that the step $y \to x$ in ρ_1 is vertical. Let $a \to c$ be the last horizontal step in ρ_1 before $y \to x$. This step exists since the paths ρ_1 and ρ_2 have the same start points and the node z in

---

[3] See further [8, Corollary 10.2.10 on page 105] and [10, Figure 4.31 on page 118].

ρ_2 lies left of the node y in ρ_1. By the construction of the reduction diagram, the node a also admits a vertical step $a \to b$ (downwards). The vertical reduction $c \twoheadrightarrow y$ that is part of ρ_1 must have been created (partially) by pasting an elementary reduction diagram into the peak $b \leftarrow a \to c$ with joining reductions $b \twoheadrightarrow d \leftarrow c$. As z lies left of y and hence left of c, this tile lies within the area surrounded by the paths ρ_1 and ρ_2. Moreover $a \to c \twoheadrightarrow d$ is part of ρ_1. □

Example 7.14. The proof of Theorem 7.13 is illustrated in the following sequence of reduction diagrams where one path is step by step converted into another path:
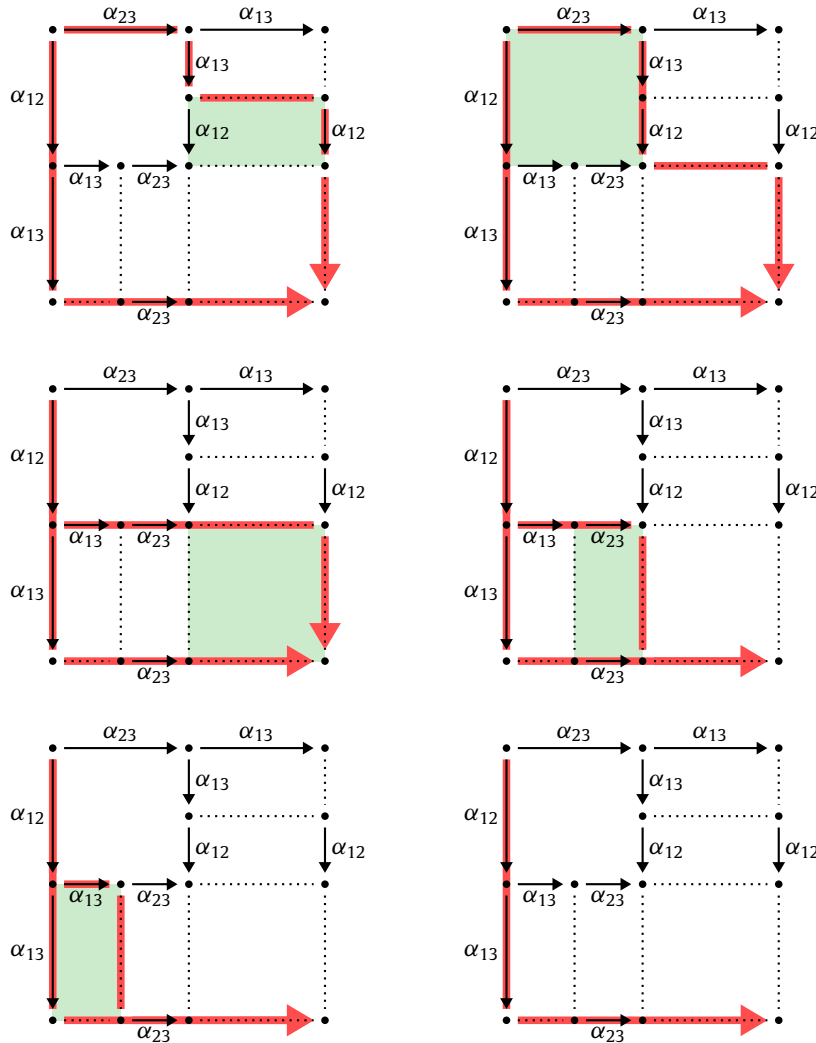
From the reduction diagrams we can extract the conversion:

$$\alpha_{23}\alpha_{13}\varepsilon\underline{\alpha_{12}}\varepsilon$$
$$\approx \underline{\alpha_{23}\alpha_{13}\alpha_{12}}\varepsilon\varepsilon$$
$$\approx \alpha_{12}\alpha_{13}\alpha_{23}\underline{\varepsilon\varepsilon}$$
$$\approx \alpha_{12}\alpha_{13}\underline{\alpha_{23}}\varepsilon\varepsilon$$
$$\approx \alpha_{12}\underline{\alpha_{13}}\varepsilon\alpha_{23}\varepsilon$$
$$\approx \alpha_{12}\alpha_{13}\varepsilon\alpha_{23}\varepsilon$$

Without the empty steps ε and trivial conversion steps, we have:

$$\alpha_{23}\alpha_{13}\alpha_{12} \approx \alpha_{12}\alpha_{13}\alpha_{23}$$

Remark 7.15. We note that extracting a conversion from a completed reduction diagram has the following benefit: it yields a *standardised* conversion where the conversion steps move from right to left. This is very much analogous to the situation in lambda calculus where also standard reductions are extracted from completed reduction diagrams [8, Chapter 9, Theorem 10.2.6 and Figures on page 100–105].

7.3. Tiling preserves simplicity

Lemma 7.16 (Simplicity is preserved under equivalence). *If v is a simple braid code and $v \approx w$, then w is simple.*

Proof. Follows immediately from the fact that Artin's equations in the braid codes notation are (special forms of) word reversal. The reversal of factors does not affect simplicity. □

Lemma 7.17 (Tiling preserves simplicity). *Let u and v be compatible simple positive braid codes. Then while tiling the reduction diagram*
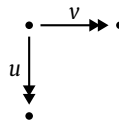

all paths in the diagram remain simple.

Proof. We prove that the following invariant holds during tiling:

($\star$) Every node in the diagram can be reached by a simple path (from the upper-left corner of the reduction diagram).

From ($\star$) it follows that every path in the reduction diagram is simple. By Theorem 7.13 all paths with the same start and end point are equivalent, and by Lemma 7.16 simplicity is preserved under equivalence. Finally, every factor of a simple path is simple. It follows that every path is simple.

Initially the invariant ($\star$) holds since the words u and v are simple. So assume that the invariant holds and we paste an elementary diagram into a peak of the form
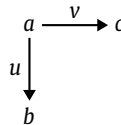

Let w be a path to a. As all paths to a, b and c are simple, it follows that wu is a simple path to b and wv is a simple path to c. We distinguish cases according to the form of the elementary diagram matching the peak:

(i) For $u = \alpha_{ij}$ and $v = \alpha_{kl}$ with pairwise distinct $i, j, k, l \in \{1, \ldots, n\}$, we get:
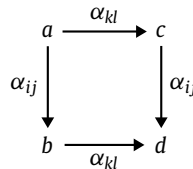
As wu and wv are simple, the path w contains none of the symbols α_{ij}, α_{ji}, α_{kl} and α_{lk}. From this it follows that $w\alpha_{ij}\alpha_{kl}$ and $w\alpha_{kl}\alpha_{ij}$ are simple again. So the invariant ($\star$) also holds for d.

(ii) For $u = \alpha_{ij}$ and $v = \alpha_{jk}$ with pairwise distinct $i, j, k \in \{1, \ldots, n\}$, we get:
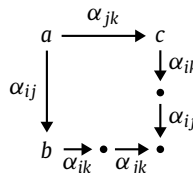
Since the node a admits steps α_{ij} and α_{jk} it follows that the strand j is between the strands i and k in a (right of i and left of k). As argued before, w contains none of the symbols α_{ij}, α_{ji}, α_{jk} and α_{kj}. So the strand i is left of j throughout w. Likewise the strand k is right of j throughout w. Thus, w cannot contain the symbols α_{ik} and α_{ki}. From this it follows that $w\alpha_{ij}\alpha_{ik}\alpha_{jk}$ and $w\alpha_{jk}\alpha_{ik}\alpha_{ij}$ are simple again. So the invariant ($\star$) is upheld.

In both cases the invariant is preserved. $\quad\square$

7.4. Termination of tiling via Newman's lemma and simple braids

We prove termination of tiling using a classical result by Newman [9] on confluence of abstract reduction systems. A relation $\rightarrow \subseteq A \times A$ is called *terminating* if there are no infinite chains

$$a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow \ldots$$

Newman's Lemma [9] states that, for a terminating relation, local confluence ($\leftarrow \cdot \rightarrow \subseteq \twoheadrightarrow \cdot \twoheadleftarrow$) implies confluence ($\twoheadleftarrow \cdot \twoheadrightarrow \subseteq \twoheadrightarrow \cdot \twoheadleftarrow$).

Lemma 7.18 (*Newman's Lemma for tiling*). *Let A be a set and $\rightarrow \subseteq A \times A$ be a terminating relation such that for every $b \leftarrow a \rightarrow c$ there exist joining reductions $b \twoheadrightarrow d \twoheadleftarrow c$ for some $d \in A$. Then $\rightarrow$ is confluent by tiling.*

We define a terminating (well-founded) relation $\succ$ on simple braid codes similar to the Bruhat order on permutations. For a word u we write

$$u^{\approx} = \{ w \mid u \approx w \}$$

for the convertibility class of u.

Lemma 7.19 (*Extension of simple words is terminating*). *Let S be the set of simple braid codes over n strands. We define $\succ \subseteq S^{\approx} \times S^{\approx}$ by $u^{\approx} \succ (u\,\alpha_{ij})^{\approx}$ for every word $u\,\alpha_{ij} \in S$. The relation $\succ$ is terminating.*

Proof. The size of the set $\{\{i,j\} \mid 1 \le i, j \le n\}$ is $\frac{n\cdot(n-1)}{2}$. Thus, for a braid with n stands, this is the maximum length of a simple braid code. Moreover, convertible braid codes have the same length. $\quad\square$

Lemma 7.20 (*Tiling simple braid codes terminates*). *Let u and v be compatible simple positive braid codes. Then tiling the reduction diagram*
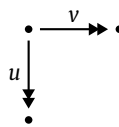


terminates.

Proof. We label the nodes in the reduction diagram as follows: if w is a path from the upper-left corner to a node x, then we label x with $w^{\approx}$. (By Theorem 7.13, all paths to x are convertible.) We use $L(x)$ to denote the label of x. By Lemma 7.16 all paths in the reduction diagram are simple. So, throughout the tiling, all nodes in the diagram have simple labels. As a consequence, whenever $x \rightarrow y$ in the reduction diagram, then $L(x) \succ L(y)$ by Lemma 7.19. So the restriction of $\rightarrow$ to the steps occurring (during tiling) in the reduction diagram is terminating. Thus by Newman's Lemma 7.18 tiling of the reduction diagram is terminating. $\quad\square$

Theorem 7.21. *Let u and v be compatible positive braid codes. Then tiling the reduction diagram*
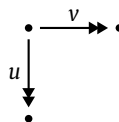
*is terminating and confluent. The unique complete tiling*
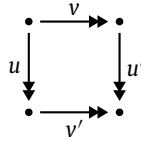
has the following properties:

  (i) *If u is simple, then so is u'. If v is simple, then so is v'.*
 (ii) *If u and v are simple, then all paths in the reduction diagram are simple (in particular, uv' and vu').*

Proof. Every word w is the concatenation of simple words (a single symbol is always simple). Thus, by induction on the length of u, it suffices to consider the case that u is simple. So let u be simple.

If v is simple, then termination follows from Lemma 7.20 and the Properties (i) and (ii) from Lemma 7.17. If v is not simple, we use induction on the length of v. For $v = \varepsilon$ there is nothing to be shown. Otherwise $v = v_1 v_2$ for words v_1, v_2 such that v_1 is simple and $v_1 \neq \varepsilon$. Then we have:
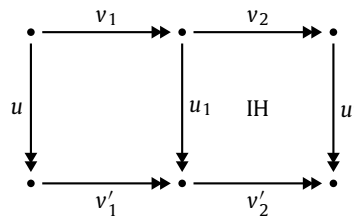

The tiling of the upper-left corner of u against v_1 terminates by Lemma 7.20. By Lemma 7.17, u_1 is simple. By the induction hypothesis (v_2 is shorter than v_1), the tiling of u_1 against v_2 is terminating and u' is simple. So we also have Property (i) for u and u' (and it follows by symmetry for v and v').

Thus tiling is terminating. Finally, confluence of the tiling process follows from Lemma 7.12. □

7.5. Termination of tiling via decreasing diagrams

We will now give an alternative proof of termination of tiling, and thereby confluence by diagram completion, using a powerful method in term rewriting called 'decreasing diagrams'. This method is based on a theorem of De Bruijn and Van Oostrom [32–34,31,35], and is the most powerful method in abstract rewriting to prove confluence by tiling with elementary diagrams. It has many well-known lemma's in abstract rewriting as corollaries, such as Newman's Lemma and Huet's Lemma.

Theorem 7.22 *(Decreasing Diagrams). Let A be a set and $\to\, \subseteq A \times A$. Then $\to$ is confluent (i.e. $\leftarrow \cdot \twoheadrightarrow\, \subseteq\, \twoheadrightarrow \cdot \twoheadleftarrow$) if the steps can be labelled with labels from a well-founded partial order $(I, <)$ such that every peak $b \overset{\beta}{\leftarrow} a \overset{\alpha}{\to} c$ can be joined by reductions of the form:*
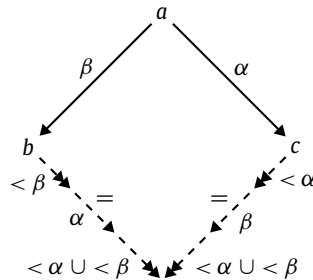
Note that α, β may or may not (=) 'cross over' to the opposite side.

We fix the number of strands n and define $\underline{n} = \{1, \ldots, n\}$. We use $\mathfrak{B}$ to denote the set of positive braid codes with n strands. On $\mathfrak{B}$ we define a relation $\to$ by

$$w \to w\,\alpha_{ij} \quad \text{for every } w\,\alpha_{ij} \in \mathfrak{B}$$

We are interested in establishing confluence of $\to$, that is, $\leftarrow \cdot \twoheadrightarrow\, \subseteq\, \twoheadrightarrow \cdot \twoheadleftarrow$. We also write $\to_{\alpha_{ij}}$ or $\overset{\alpha_{ij}}{\to}$ for a step $w \to w\,\alpha_{ij}$. So $\to\, = \bigcup_{i,j\,\in\,\underline{n}} \to_{\alpha_{ij}}$.

Let $perm(w) : \underline{n} \to \underline{n}$ be the permutation arising from a code $w \in \mathfrak{B}$. So $perm(w)(i)$ is the position of the i-th strand after w. We write $id_{\underline{n}}$ for the identity function on $\underline{n}$. For a permutation $p : \underline{n} \to \underline{n}$ we define

$$order(p) = \{ \langle i, j \rangle \mid p(i) < p(j) \}$$

We have $|order(p)| = n \cdot (n-1)/2$.

Example 7.23. Let $n = 4$ and define $p : \underline{n} \to \underline{n}$ by

$$\begin{pmatrix} 1234 \\ 3124 \end{pmatrix}$$

In the sequel we will denote a permutation by the image tuple, 3124 in this case. Then $order(p) = \{ \langle 3, 1 \rangle, \langle 3, 2 \rangle, \langle 3, 4 \rangle, \langle 1, 2 \rangle, \langle 1, 4 \rangle, \langle 2, 4 \rangle \}$.

We define $moves : \mathfrak{B} \to \mathcal{P}(\underline{n} \times \underline{n})$ as follows:

$$moves(\varepsilon) = \varnothing$$

$$moves(w\, \alpha_{ij}) = \begin{cases} moves(w) \setminus \{ \langle i, j \rangle \}, & \text{if } \langle i, j \rangle \in moves(w) \\ order(perm(w)) \setminus \{ \langle i, j \rangle \}, & \text{if } \langle i, j \rangle \notin moves(w) \end{cases}$$

Lemma 7.24. *For every $w \in \mathfrak{B}$ we have: $moves(w) \subseteq order(perm(w))$ and $moves(w)$ is transitive.*

Proof. Clearly, the claimed properties hold for $moves(\varepsilon)$. It suffices to show that the properties are invariant under extension of braid codes. Let $w\, \alpha_{ij} \in \mathfrak{B}$ such that $moves(w) \subseteq order(perm(w))$ and $moves(w)$ is transitive. We have

$$order(perm(w\, \alpha_{ij})) = (order(perm(w)) \setminus \{ \langle i, j \rangle \}) \cup \{ \langle j, i \rangle \} \tag{5}$$

We distinguish cases:

(i) For $\langle i, j \rangle \in moves(w)$ we have:

$$moves(w\, \alpha_{ij}) = moves(w) \setminus \{ \langle i, j \rangle \} \subseteq order(perm(w)) \setminus \{ \langle i, j \rangle \}$$

(ii) For $\langle i, j \rangle \notin moves(w)$ we have:

$$moves(w\, \alpha_{ij}) = order(perm(w)) \setminus \{ \langle i, j \rangle \}$$

In both cases $moves(w\, \alpha_{ij}) = order(perm(w)) \setminus \{ \langle i, j \rangle \} \subseteq order(perm(w\, \alpha_{ij}))$ by (5).

Transitivity of $moves(w\, \alpha_{ij})$ can be seen as follows. Let k, l, m such that $\langle k, l \rangle, \langle l, m \rangle \in moves(w\, \alpha_{ij})$. From $moves(w\, \alpha_{ij}) \subseteq order(perm(w))$ it follows that the strand l is between the strands k and m after w. Thus the strands k and m are not adjacent after w. Thus $\langle k, m \rangle \neq \langle i, j \rangle$ since $w\, \alpha_{ij} \in \mathfrak{B}$ and only adjacent strands can be swapped. We again distinguish cases:

(i) For $\langle i, j \rangle \in moves(w)$ we have $\langle k, l \rangle, \langle l, m \rangle \in moves(w)$ as a consequence of $moves(w\, \alpha_{ij}) \subseteq moves(w)$. Then $\langle k, m \rangle \in moves(w)$ by transitivity of $moves(w)$. Thus $\langle k, m \rangle \in moves(w\, \alpha_{ij})$ since $\langle k, m \rangle \neq \langle i, j \rangle$.
(ii) For $\langle i, j \rangle \notin moves(w\, \alpha_{ij})$ we have $\langle k, l \rangle, \langle l, m \rangle \in order(perm(w))$. Then $\langle k, m \rangle \in order(perm(w))$ by transitivity of $order(perm(w))$, and $\langle k, m \rangle \in moves(w\, \alpha_{ij})$ since $\langle k, m \rangle \neq \langle i, j \rangle$.

Thus $moves(w\, \alpha_{ij})$ is transitive. $\square$

We define the *height* of $w \in \mathfrak{B}$ by

$$height(w) = |moves(w)| \,.$$

We show that all elementary diagrams for braids are decreasing when the steps are labelled with the height of their target.

Definition 7.25 *(Good and bad steps).* We say that a step $u \overset{\alpha_{ij}}{\to} w$ is *good* if $\langle i, j \rangle \in moves(u)$ and *bad*, otherwise.

Basically, the intuition is ensuing from the fact that a braid code can be cut into simple factors; when leaving such a simple part apparently something 'drastic' happens and this is a bad step.
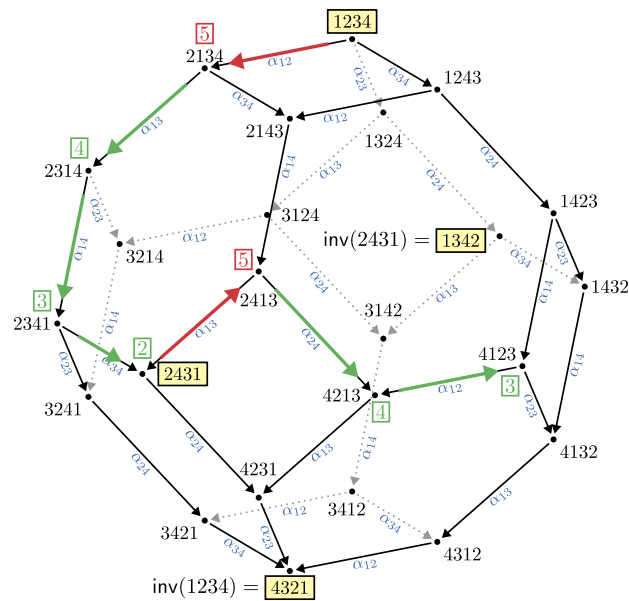
Fig. 8. Re-adjustment of travel goals during travel over P_4. Residing at 1234 the traveller decides not to stay but starts travelling to 4321, the antipodal point of 1234. Each step in the direction of the goal 4321 the amount of travel still to do decreases by 1. Arriving in 2431, the traveller veeres off from reaching the goal 4321, and heads for a new goal, the antipodal point of 2431 as the next goal, 1342. This is a red (bad) step, increasing the steps to do to 5 again. Two more green (good) steps are done towards 1342. Not reaching that goal, the travel stops in 4123.

Example 7.26. Let $n = 4$ and consider the code $w = \alpha_{12}\alpha_{13}\alpha_{14}\alpha_{34}\alpha_{31}\alpha_{24}\alpha_{21}$. Then

braid code	perm	moves	height	good/bad
ε	1234		0	
α_{12}	2134	13, 14, 23, 24, 34	5	bad
$\alpha_{12}\alpha_{13}$	2314	14, 23, 24, 34	4	good
$\alpha_{12}\alpha_{13}\alpha_{14}$	2341	23, 24, 34	3	good
$\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{34}$	2431	23, 24	2	good
$\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{34}\alpha_{31}$	2413	24, 23, 21, 43, 41	5	bad
$\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{34}\alpha_{31}\alpha_{24}$	4213	23, 21, 43, 41	4	good
$\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{34}\alpha_{31}\alpha_{24}\alpha_{21}$	4213	23, 43, 41	3	good

In the *moves* column we write ij as shorthand for $\langle i, j \rangle$. The travel over P_4 corresponding to this word is displayed in Fig. 8. The upshot is that the word is indexed with heights as follows

$$w = {}_0\alpha_{12}{}_5\alpha_{13}{}_4\alpha_{14}{}_3\alpha_{34}{}_2 \mid \alpha_{31}{}_5\alpha_{24}{}_4\alpha_{21}{}_3 .$$

Let $H = n \cdot (n-1)/2 - 1$. The following two lemmas are immediate:

Lemma 7.27. *For every $w \in \mathfrak{B}$ we have $0 \leq \mathrm{height}(w) \leq H$.*

Lemma 7.28. *For a step $u \to w$ we have $\mathrm{height}(u) > \mathrm{height}(w)$ if the step is good, and $\mathrm{height}(w) = H$ if and only if the step is bad.*

Lemma 7.29. *For $w \in \mathfrak{B}$ and a simple braid code $\alpha_{i_1 j_1} \cdots \alpha_{i_m j_m}$, the sequence*

$$w_1 \to_{\alpha_{i_1 j_1}} w_2 \to_{\alpha_{i_2 j_2}} \cdots \to_{\alpha_{i_m j_m}} w_{m+1}$$

contains at most 1 bad step.

Proof. It suffices to consider the case that $w_1 \to_{\alpha_{i_1 j_1}} w_2$ is a bad step. Since the step $w_1 \to_{\alpha_{i_1 j_1}} w_2$ is bad, we have

$$moves(w_2) = order(perm(w_1)) \setminus \{ \langle i_1, j_1 \rangle \}$$

J. Endrullis, J.W. Klop / Theoretical Computer Science ••• (••••) •••–•••

As the braid code $\alpha_{i_1 j_1} \cdots \alpha_{i_m j_m}$ is simple we have $perm(w_1)(i_k) < perm(w_1)(j_k)$ and thus $\langle i_k, j_k \rangle \in order(perm(w_1))$ for every $k \in \{1, \ldots, m\}$. It follows that $\langle i_k, j_k \rangle \in moves(w_k)$ for every $k \in \{2, \ldots, m\}$. Hence the remaining steps in the sequence are good. □

Theorem 7.30. *Extension of positive braid codes is confluent by tiling.*

Proof. We show that braid extension $\to$ on positive braid codes $\mathfrak{B}$ is confluent by tiling. For this purpose, we use decreasing diagrams where we tacitly label steps $x \to y$ by height $height(y)$ of their target. So, when we speak of the *height of a step* $x \to y$ we refer to $height(y)$. For confluence by tiling of $\to$ it suffices to show that every peak $b \leftarrow a \to c$ can be joined by decreasing elementary diagrams as in Theorem 7.22.

So consider a peak $b \leftarrow a \to c$. This peak can be joined by one of the elementary diagrams in Definition 7.2 by Lemma 7.11, say
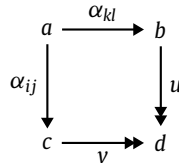
for some $i, j, k, l \in \underline{n}$. For each elementary diagram in Definition 7.2, $\alpha_{ij} v$ and $\alpha_{kl} u$ are simple words. Thus by Lemma 7.29
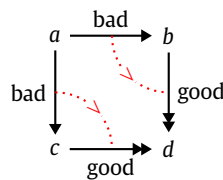
(i) the reduction $a \xrightarrow{\alpha_{ij}} c \xrightarrow{v} d$ contains at most one bad step, and
(ii) the reduction $a \xrightarrow{\alpha_{kl}} b \xrightarrow{u} d$ contains at most one bad step.

By Lemmas 7.27 and 7.28 we have:

(iii) a bad step is greater than a good step, and
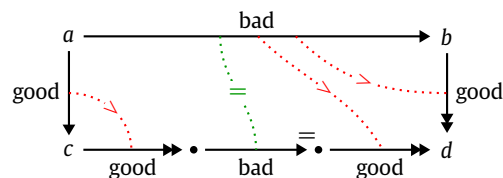(iv) a good step is smaller than every directly preceding step.

We distinguish cases:

(a) If $a \xrightarrow{\alpha_{ij}} c$ and $a \xrightarrow{\alpha_{kl}} b$ are bad steps, then by (i), (ii) and (iii) we have:

This elementary diagram is decreasing.

(b) If $a \xrightarrow{\alpha_{ij}} c$ is good and $a \xrightarrow{\alpha_{kl}} b$ is bad, then by (i)–(iv) we have:


This elementary diagram is decreasing. The decreasing height of the steps is indicated by the dotted lines inside the diagrams.

(c) The case $a \xrightarrow{\alpha_{ij}} c$ is bad and $a \xrightarrow{\alpha_{kl}} b$ is good, is symmetric to (b).

(d) Finally, assume that both $a \xrightarrow{\alpha_{ij}} c$ and $a \xrightarrow{\alpha_{kl}} b$ are good.
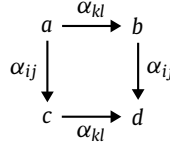   Then we have $\langle i, j \rangle, \langle k, l \rangle \in moves(a)$. We show that then the joining reduction do not contain bad steps. Then we have a decreasing elementary diagram of the form
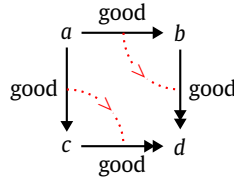
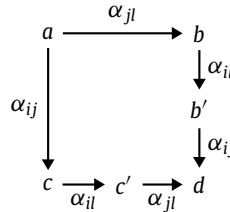It suffices to consider the non-trivial elementary diagrams:

- If $\{i,j\} \cap \{k,l\} = \varnothing$, then

From $\langle i,j \rangle, \langle k,l \rangle \in moves(a)$ it follows that $\langle i,j \rangle \in moves(b)$ and $\langle k,l \rangle \in moves(c)$. Thus the joining steps are good.

- If $j = k$ (or symmetrically $i = l$), then


From $\langle i,j \rangle, \langle j,l \rangle \in moves(a)$ it follows that $\langle i,l \rangle \in moves(a)$ by Lemma 7.24. Then $\langle i,l \rangle, \langle i,j \rangle \in moves(b)$ and $\langle i,j \rangle \in moves(b')$. Likewise $\langle i,l \rangle, \langle j,l \rangle \in moves(c)$ and $\langle j,l \rangle \in moves(c')$ Thus all joining steps are good.

- If $i = k$ or $j = l$ we have a trivial decreasing elementary diagram with empty joining sides.

Every peak can be joined by a decreasing elementary diagram. Hence by Theorem 7.22, the relation $\rightarrow$ is confluent by tiling. $\square$

Example 7.31. In this example we write overline $\overline{w}$ for infinite repetition of w. We do not consider infinite braid codes, but the cyclic braid code can be cut-off at any desired point. We write r (red) for bad steps, and g (green) for good steps.

(i) The cyclic path $\overline{\alpha_{12}\alpha_{21}} = \overline{rr} = \overline{55}$ has only bad steps of height 5.
(ii) The cycle $\overline{\alpha_{12}\alpha_{34}\alpha_{21}\alpha_{43}} = \overline{rgrg} = \overline{5454}$ alternates between bad and good steps of height 5 and 4, respectively.
(iii) The word $\alpha_{12}\overline{\alpha_{13}\alpha_{14}\alpha_{34}} = r\overline{gggrrr} = 5\overline{432555}$, first α not overlined, then the next three overlined.
(iv) Let $w = \alpha_{12}\alpha_{13}\alpha_{14}\alpha_{34}\alpha_{13}\alpha_{24} = rgggrg$, and $v = \alpha_{12}\alpha_{13}\alpha_{34}\alpha_{13}\alpha_{41} = rgggr$. These two braid codes w, v against each other give a reduction diagram containing, after the common prefix of the words have cancelled out each other, an elementary diagram which is decreasing, and has a red step crossing over to the other side:
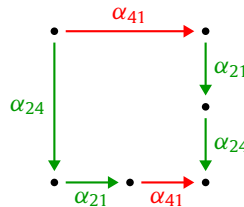
Note that decreasing elementary diagrams with double cross-over are not possible.

7.6. Reduction diagrams and projection equivalence

We will use completed reduction diagrams to connect projection equivalence $\simeq$ with convertibility $\approx$ of positive braid codes. We follow here a development that is classic in lambda calculus and orthogonal rewriting theory, and has its origins in the work of Lévy [5] for lambda calculus and of Huet and Lévy [4] for orthogonal rewriting.
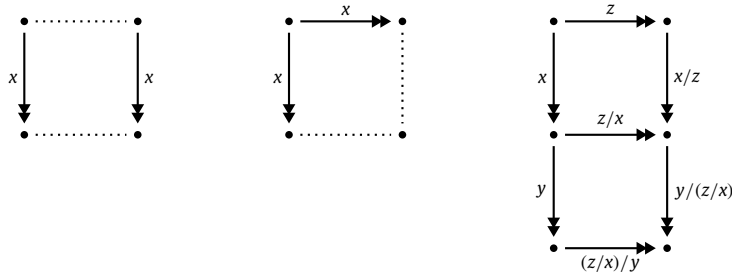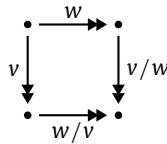
Here is the basic definition.

Fig. 9. Lévy's reduction diagram equations.

Definition 7.32 *(Projection).* Let w, v be compatible positive braid codes. Then w/v is bottom side of the completed reduction diagram setting w horizontally (top side) against v vertically (left side). Pictorially that is:

Definition 7.33 *(Projection order).* For compatible braid codes v, w we write $v \sqsubseteq w$ if $v/w = \varepsilon$.

Definition 7.34 *(Projection equivalence).* Two compatible braid codes v, w are *projection equivalent* (or *Lévy equivalent*), denoted $v \simeq w$, if $v \sqsubseteq w$ and $w \sqsubseteq v$.

Lévy established the following useful equations for projection:

$$x \cdot \varepsilon = x$$
$$\varepsilon \cdot x = x$$
$$x/\varepsilon = x$$
$$\varepsilon/x = \varepsilon$$
$$x/x = \varepsilon$$
$$xy/z = (x/z)(y/(z/x))$$
$$z/xy = (z/x)/y$$

They are instrumental for deconstructing reduction diagrams, as sometimes necessary in inductive proofs concerning projections and projection equivalence. The proof is immediate from the definition and Fig. 9. They are also used in the literature on braid theory, see Section 8.

Lemma 7.35 *(Projection equivalence implies convertibility).* If $w \simeq v$ then $w \approx v$.

Proof. We have $w(v/w) \approx v(w/v)$ by Theorem 7.13. From the Lévy equivalence we get $v/w = w/v = \varepsilon$, so $w \approx v$. $\square$

In order to prove the much more difficult reverse implication, we now establish an important congruence property of convertibility with respect to projection.

Lévy diagram equations can also be applied fruitfully to give a complete (terminating and confluent) term rewriting system for projection, where word concatenation is associative:

$$\alpha_{ij}/\alpha_{kl} \to \alpha_{ij} \qquad \text{for pairwise distinct } i, j, k, l \in \{1, \ldots, n\}$$
$$\alpha_{ij}/\alpha_{jk} \to \alpha_{ik}\alpha_{jk} \qquad \text{for pairwise distinct } i, j, k \in \{1, \ldots, n\}$$
$$x\varepsilon \to x$$
$$\varepsilon x \to x$$
$$x/\varepsilon \to x$$

$$\varepsilon / x \to \varepsilon$$

$$x / x \to \varepsilon$$

$$xy / z \to (x/z)(y/(z/x))$$

$$z / xy \to (z/x)/y$$

This is basically a reformulation of completing reduction diagrams by tiling with elementary diagrams which has been proved above to be terminating and confluent.

7.7. Congruence with respect to projection

We prove the following congruence property for projection.

Theorem 7.36. *Let u_1, v_1 be compatible positive braid codes. If $u_1 \approx u_2$ and $v_1 \approx v_2$, then $u_1/v_1 \approx u_2/v_2$ and $v_1/u_1 \approx v_2/u_2$. In other words: conversion is a congruence with respect to the projection operator.*

This subsection is devoted to the proof of this theorem. The proof proceeds in several steps. First, we prove that the theorem holds for u a single letter and $v_1 \approx v_2$ an instance of one of Artin's equations for braid codes (Lemmas 7.37 and 7.38). Next, we lift this result to simple positive braid codes by induction with respect to the Bruhat order. Finally, we generalise the result to general positive braid codes by 'cutting' the braid codes into simple parts.

Lemma 7.37. *For $p, q, i, j, k, l \in \{1, \ldots, n\}$ such that i, j, k, l are pairwise distinct, it holds that*

(i) $\alpha_{pq}/\alpha_{ij}\alpha_{kl} \approx \alpha_{pq}/\alpha_{kl}\alpha_{ij}$, *and*
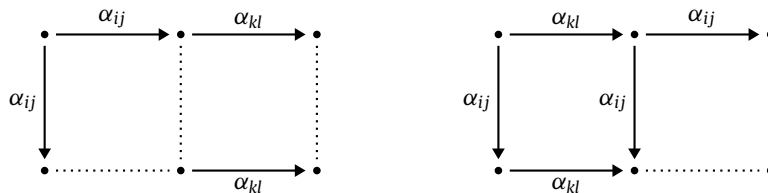(ii) $\alpha_{ij}\alpha_{kl}/\alpha_{pq} \approx \alpha_{kl}\alpha_{ij}/\alpha_{pq}$.

Proof. By symmetry of Artin's equation for braid codes $\alpha_{ij}\alpha_{kl} \approx \alpha_{kl}\alpha_{ij}$, we may assume that $i, j < k, l$. We distinguish the following cases:

(i) If $\{p, q\} \cap \{i, j, k, l\} = \varnothing$, then $\alpha_{pq}/\alpha_{ij}\alpha_{kl} = \alpha_{pq} = \alpha_{pq}/\alpha_{kl}\alpha_{ij}$ and $\alpha_{ij}\alpha_{kl}/\alpha_{pq} = \alpha_{ij}\alpha_{kl} \approx \alpha_{kl}\alpha_{ij} = \alpha_{kl}\alpha_{ij}/\alpha_{pq}$.
(ii) If $p < i$ and $q = i$, then:
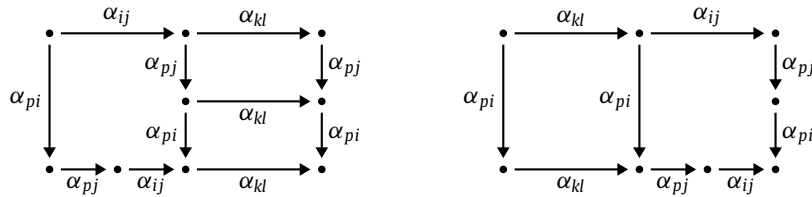
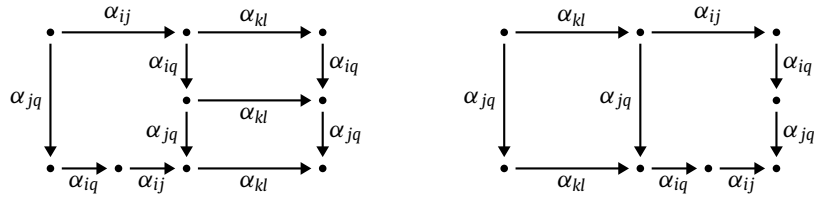Here $\alpha_{pj}\underline{\alpha_{ij}\alpha_{kl}} \approx \underline{\alpha_{pj}\alpha_{kl}}\alpha_{ij} \approx \alpha_{kl}\alpha_{pj}\alpha_{ij}$.
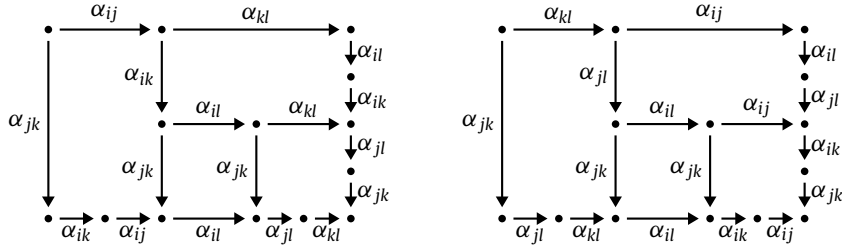(iii) If $p = i$ and $q = j$, then:

(iv) If $p = j$ and $q < k$, then:

Here $\alpha_{iq}\underline{\alpha_{ij}\alpha_{kl}} \approx \alpha_{iq}\underline{\alpha_{kl}\alpha_{ij}} \approx \alpha_{kl}\alpha_{iq}\alpha_{ij}$.

(v) If $p = j$ and $q = k$, then:

Here we have

$$\alpha_{il}\underline{\alpha_{ik}\alpha_{jl}}\alpha_{jk} \approx \alpha_{il}\alpha_{jl}\alpha_{ik}\alpha_{jk}$$

and

$$\alpha_{ik}\underline{\alpha_{ij}\alpha_{il}\alpha_{jl}}\alpha_{kl}$$

$$\approx \alpha_{ik}\underline{\alpha_{jl}\alpha_{il}}\alpha_{ij}\alpha_{kl}$$

$$\approx \alpha_{jl}\alpha_{ik}\alpha_{il}\underline{\alpha_{ij}\alpha_{kl}}$$

$$\approx \alpha_{jl}\underline{\alpha_{ik}\alpha_{il}\alpha_{kl}}\alpha_{ij}$$

$$\approx \alpha_{jl}\alpha_{kl}\alpha_{il}\alpha_{ik}\alpha_{ij}$$

(vi) The case $p = k$ and $q = l$ is symmetric to the case $p = i$ and $q = j$.

(vii) The case $p = l$ and $q > l$ is symmetric to the case $p < i$ and $q = i$.

The case distinction is exhaustive and we have $\alpha_{pq}/\alpha_{ij}\alpha_{kl} \approx \alpha_{pq}/\alpha_{kl}\alpha_{ij}$, and $\alpha_{ij}\alpha_{kl}/\alpha_{pq} \approx \alpha_{kl}\alpha_{ij}/\alpha_{pq}$ for each case. So we have proven the claim. □

Lemma 7.38. *For $p, q, i, j, k \in \{1, \ldots, n\}$ such that i, j, k are pairwise distinct, it holds that*

(i) $\alpha_{pq}/\alpha_{ij}\alpha_{ik}\alpha_{jk} \approx \alpha_{pq}/\alpha_{jk}\alpha_{ik}\alpha_{ij}$, *and*

(ii) $\alpha_{ij}\alpha_{ik}\alpha_{jk}/\alpha_{pq} \approx \alpha_{jk}\alpha_{ik}\alpha_{ij}/\alpha_{pq}$.

Proof. By symmetry of Artin's equation for braid codes $\alpha_{ij}\alpha_{ik}\alpha_{jk} \approx \alpha_{jk}\alpha_{ik}\alpha_{ij}$, we may assume that $i < j < k$. We distinguish the following cases:
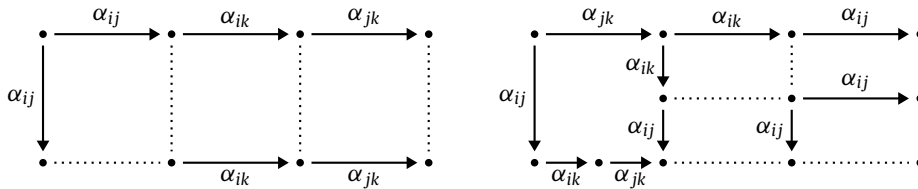
(i) If $\{p, q\} \cap \{i, j, k\} = \varnothing$, then $\alpha_{pq}/\alpha_{ij}\alpha_{ik}\alpha_{jk} = \alpha_{pq} = \alpha_{pq}/\alpha_{jk}\alpha_{ik}\alpha_{ij}$ and $\alpha_{ij}\alpha_{ik}\alpha_{jk}/\alpha_{pq} = \alpha_{ij}\alpha_{ik}\alpha_{jk} \approx \alpha_{jk}\alpha_{ik}\alpha_{ij} = \alpha_{jk}\alpha_{ik}\alpha_{ij}/\alpha_{pq}$.

(ii) If $p < i$ and $q = i$, then:
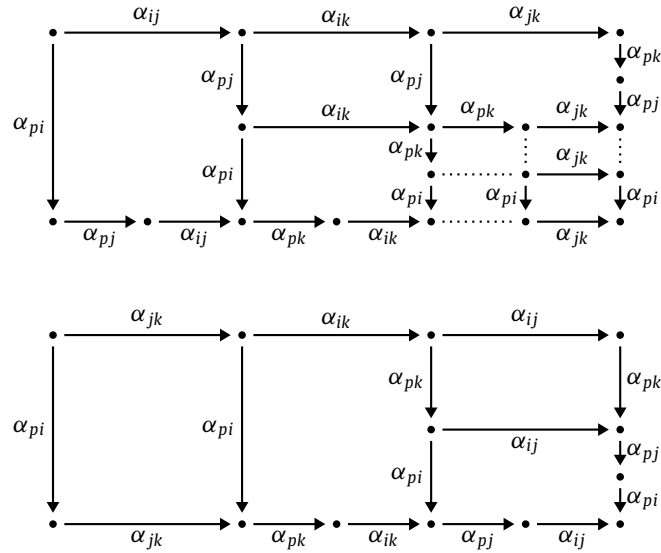
Here the right sides are literally equal and for the bottom sides we have

$$\alpha_{jk}\alpha_{pk}\underline{\alpha_{ik}\alpha_{pj}}\alpha_{ij}$$
$$\approx \underline{\alpha_{jk}\alpha_{pk}\alpha_{pj}}\alpha_{ik}\alpha_{ij}$$
$$\approx \alpha_{pk}\alpha_{pj}\underline{\alpha_{jk}\alpha_{ik}\alpha_{ij}}$$
$$\approx \alpha_{pk}\underline{\alpha_{pj}\alpha_{ik}}\alpha_{ij}\alpha_{jk}$$
$$\approx \alpha_{pk}\underline{\alpha_{ik}\alpha_{pj}}\alpha_{ij}\alpha_{jk}$$

(iii) If $p = i$ and $q = j$, then:

(iv) The case $p = j$ and $q = k$ is symmetric to $p = i$ and $q = j$.
(v) The case $p = k$ and $q > k$ is symmetric to $p < i$ and $q = i$.

The case distinction is exhaustive, and for each case we have established that $\alpha_{pq}/\alpha_{ij}\alpha_{ik}\alpha_{jk} \approx \alpha_{pq}/\alpha_{jk}\alpha_{ik}\alpha_{ij}$, and $\alpha_{ij}\alpha_{ik}\alpha_{jk}/\alpha_{pq} \approx \alpha_{jk}\alpha_{ik}\alpha_{ij}/\alpha_{pq}$. So the lemma is proven. □

We have shown that Theorem 7.36 holds for single steps projected over Artin's equations.
We first prove the Theorem 7.36 for simple braid codes.

Proof of Theorem 7.36 for simple braid codes. We prove

($\star$) For all compatible simple braid codes u_1, u_2, v_1, v_2 with $u_1 \approx u_2$ and $v_1 \approx v_2$, it holds that $u_1/v_1 \approx u_2/v_2$ and $v_1/u_1 \approx v_2/u_2$.

We tacitly label the nodes in the reduction diagrams by convertibility classes of simple braid codes as follows. We write $L(x)$ for the label of the node x. The upper-left corner of a reduction diagram can be labelled with the convertibility class of any simple word. Whenever there is a step $x \to_{\alpha_{ij}} y$ in the reduction diagram and $L(x) = w^\approx$, then we let $L(y) = (w\,\alpha_{ij})^\approx$. Roughly speaking, every node in the diagram is labelled with the label of the upper-left corner concatenated with the path to the node. As labels we only allow (equivalence classes of) simple words; you may also think of these labels as points on

the permutohedron P_n subject to the terminating Bruhat order. By Lemma 7.17 every reduction diagram for simple words can be fully labelled in this way; we can label the upper left corner with $\varepsilon^\approx$ and all path in the diagram are simple.

We prove $(\star)$ by well-founded induction with respect to the label of the upper-left corner of the reduction diagram in the order $\succ$ (Lemma 7.19).

Employing a nested induction on the lengths of the conversions $u_1 \approx u_2$ and $v_1 \approx v_2$ with respect to Artin's equations, it suffices to consider the case of a single conversion step (longer conversions can be simulated by repeatedly applying single steps). By symmetry we may assume that $u_1 = u_2$, $v_1 = xy_1z$ and $v_2 = xy_2z$ where either

(i) $y_1 = \alpha_{ij}\alpha_{kl}$ and $y_2 = \alpha_{kl}\alpha_{ij}$, or
(ii) $y_1 = \alpha_{ij}\alpha_{ik}\alpha_{jk}$ and $y_2 = \alpha_{jk}\alpha_{ik}\alpha_{ij}$,

for all pairwise distinct $i, j, k, l \in \{1, \ldots, n\}$. Let $u = u_1 = u_2$.

For $x \neq \varepsilon$, the claim follows from inspection of the following diagram:
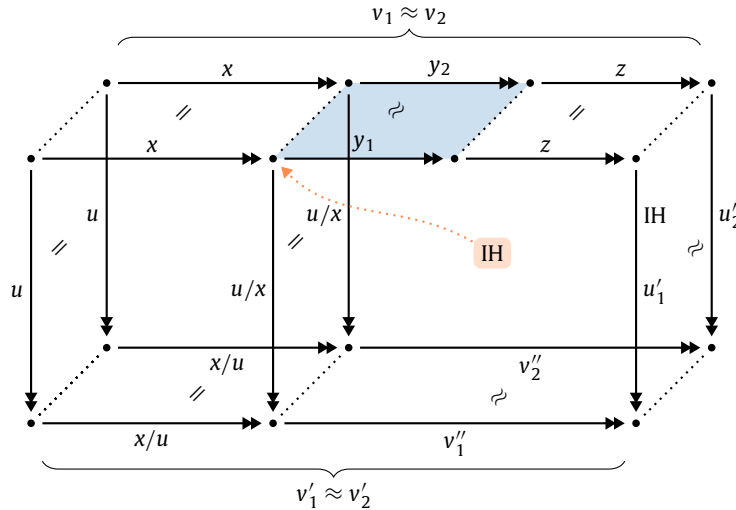
By induction hypothesis (IH) it holds that $u_1' \approx u_2'$ and $v_1'' \approx v_2''$. Thus also $v_1' \approx v_2'$ where $v_1' = (x/u)v_1''$ and $v_2' = (x/u)v_2''$.

For $x = \varepsilon$, we consider the first symbol u, say $u = \alpha_{pq}w$. Then the picture is as follows
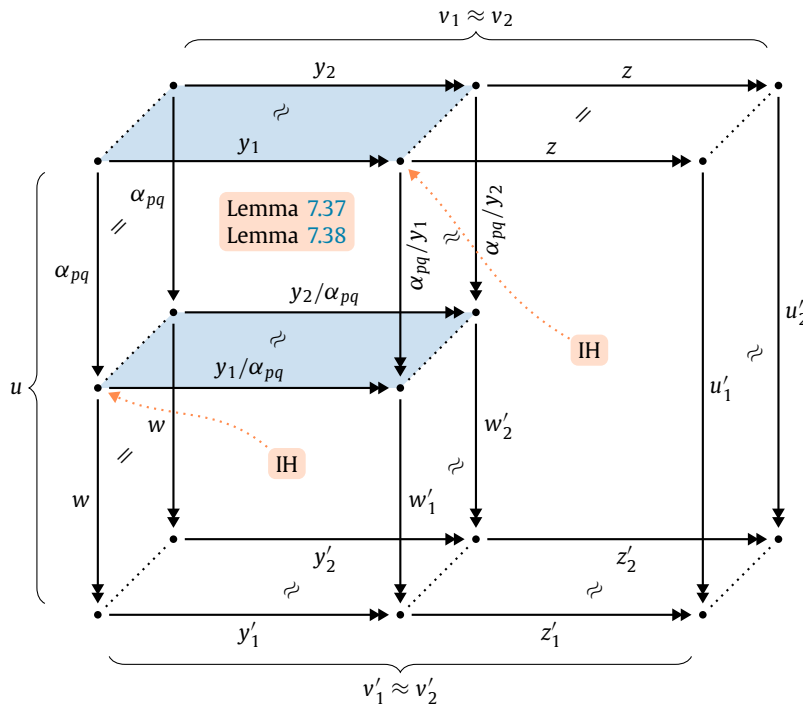
We have $y_1/\alpha_{pq} \approx y_2/\alpha_{pq}$ and $\alpha_{pq}/y_1 \approx \alpha_{pq}/y_2$ for (i) and (ii) by Lemmas 7.37 and 7.38, respectively. Then $y_1' \approx y_2'$ and $w_1' \approx w_2'$ by the induction hypothesis, and hence $z_1' \approx z_2'$ and $u_1' \approx u_2'$ by another invocation of the induction hypothesis.   □

We lift the result from simple braid codes to general positive braid codes.

Proof of Theorem 7.36. We have seen

($\star$) Theorem 7.36 holds for simple braid codes.

Now we lift this result to general positive braid codes. By induction on the lengths of the conversions $u_1 \approx u_2$ and $v_1 \approx v_2$ with respect to Artin's equations, it suffices to consider the case of a single conversion step. Then there exist natural numbers k, l and simple words $u_{1,1}, \ldots, u_{1,k}, u_{2,1}, \ldots, u_{2,k}, v_{1,1}, \ldots, v_{1,l}$ and $v_{2,1}, \ldots, v_{2,l}$ such that

$$u_1 = u_{1,1}u_{1,2}\cdots u_{1,k} \qquad v_1 = v_{1,1}v_{1,2}\cdots v_{1,l}$$

$$u_2 = u_{2,1}u_{2,2}\cdots u_{2,k} \qquad v_2 = v_{2,1}v_{2,2}\cdots v_{2,l}$$

$$\forall i.\, u_{1,i} \approx u_{2,i} \qquad\qquad \forall i.\, v_{1,i} \approx v_{2,i}$$

That is, we cut u_1, u_2, v_1, v_2 into simple braid codes. Here we use the assumption that there is only one conversion step. For longer conversions it is not guaranteed that we can cut in a way that the conversions fall into the simple parts (and do not go across parts). As there is only one conversion step, all but one of the $u_{1,i} \approx u_{2,i}$ and $v_{1,i} \approx v_{2,i}$ will be equalities $=$.

So we have the picture as shown in Fig. 10. This diagram can be fully tiled using ($\star$) since each peak (left)
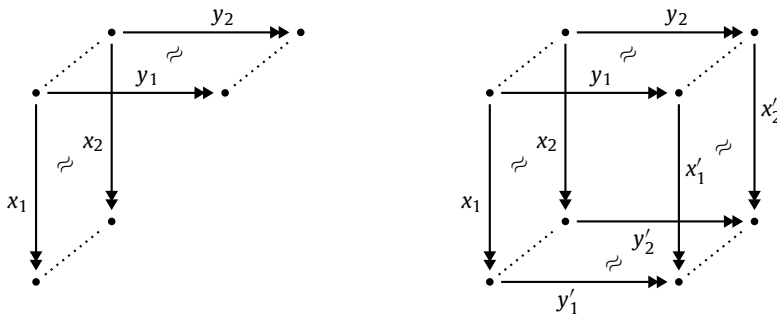
for simple braid codes x_1, x_2, y_1, y_2 can be joined as shown on the right with simple words x_1', x_2', y_1', y_2'. Hence $u_1/v_1 \approx u_2/v_2$ and $v_1/u_1 \approx v_2/u_2$.   □

7.8. More basic properties

We will now collect the harvest of Theorem 7.36.

Theorem 7.39 (*Projection equivalence is convertibility*). *Let u, v be compatible positive braid codes. Then $u \approx v$ if and only if $u \simeq v$.*

Proof. The direction '$\Leftarrow$' is Lemma 7.35. We consider the direction '$\Rightarrow$'. We have $u/u = \varepsilon$. So by Theorem 7.36 it follows that $v/u \approx \varepsilon$ and $u/v \approx \varepsilon$. Hence $v/u = u/v = \varepsilon$.   □

So we have verified that projection equivalence is indeed an equivalence.

Theorem 7.40 (*Tiling is optimal*). *Let u and v be compatible positive braid codes. Let u' be a shortest word such that $uv' \approx vu'$ for some word v'. Then we have that $u' \approx u/v$ and $v' \approx v/u$.*
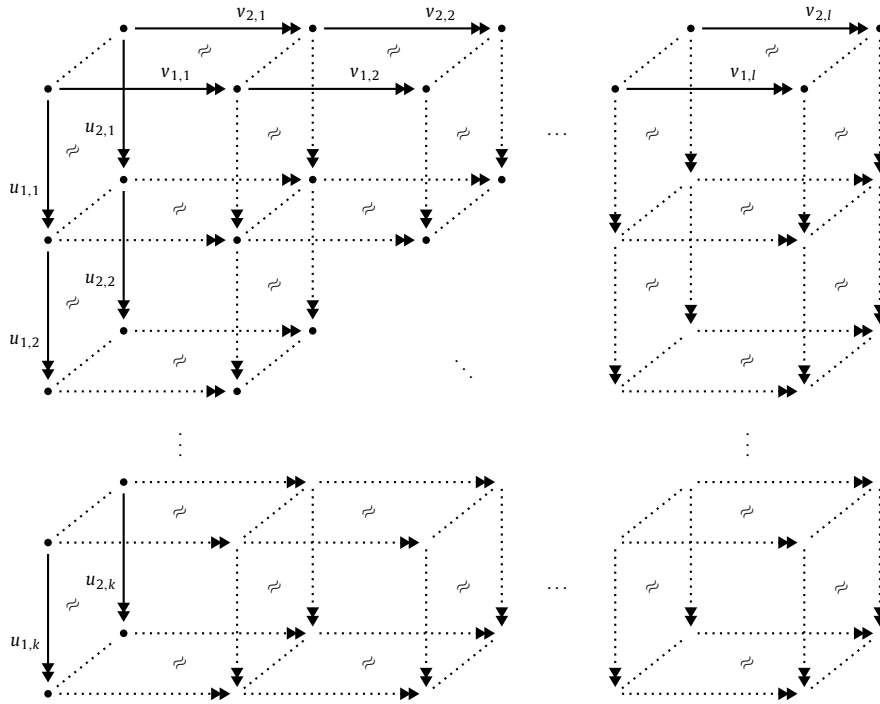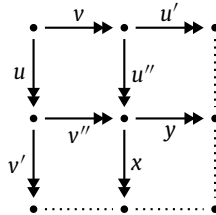
Fig. 10. Congruence with respect to projection: from simple to general braid codes.

Proof. Let u, v, u' and v' as in the theorem. So $vu' \approx uv'$. Let $u'' = u/v$ and $v'' = v/u$. The property $uv'' \approx vu''$ holds by Theorem 7.13. By choice of u' we have $|u'| \leq |u''|$. Consider the reduction diagram

We know that $uv'/vu' = vu'/uv' = \varepsilon$ by Corollary 7.42. Moreover $u' \approx u''y$ and $v' \approx v''x$ by Theorem 7.13. This implies that $|u''| \leq |u'|$ and $|v''| \leq |v'|$. Thus $|u''| = |u'|$, $|v''| = |v'|$ and $x = y = \varepsilon$. Hence $u' \approx u'' = u/v$ and $v' \approx v'' = v/u$. $\square$

Thus tiling is optimal in the sense that it gives the shortest confluent extensions.

Remark 7.41. The original ingenious confluence proof by Garside provides much longer solutions to confluence: for given u, v, the solution x, y such that $ux \approx vy$ has one of x, y equal to Δ^m with $m = |x|$.

Corollary 7.42. *Let u, v be compatible positive braid codes. Then $u \sqsubseteq v$ if and only if $uv' \approx v$ for some braid code v'.*

Proof. The direction '$\Rightarrow$' follows from Lemma 7.35. For '$\Leftarrow$', let $uv' \approx v$ for some word v'. Then $u/v \approx \varepsilon$ by Theorem 7.40. So $u \sqsubseteq v$. $\square$

Corollary 7.43 *(Cube identity). Let u, v, w be positive braid codes. Then*

$$(u/v)/(w/v) \approx (u/w)/(v/w) .$$

Proof. From Theorem 7.13 it follows that $v(w/v) \approx w(v/w)$ as these are two paths from the upper-left corner to the lower-right corner in the reduction diagram setting v against w. By Theorem 7.36 we obtain

$$(u/v)/(w/v) = u/(v(w/v)) \approx u/(w(v/w)) = (u/w)/(v/w) . \quad \square$$
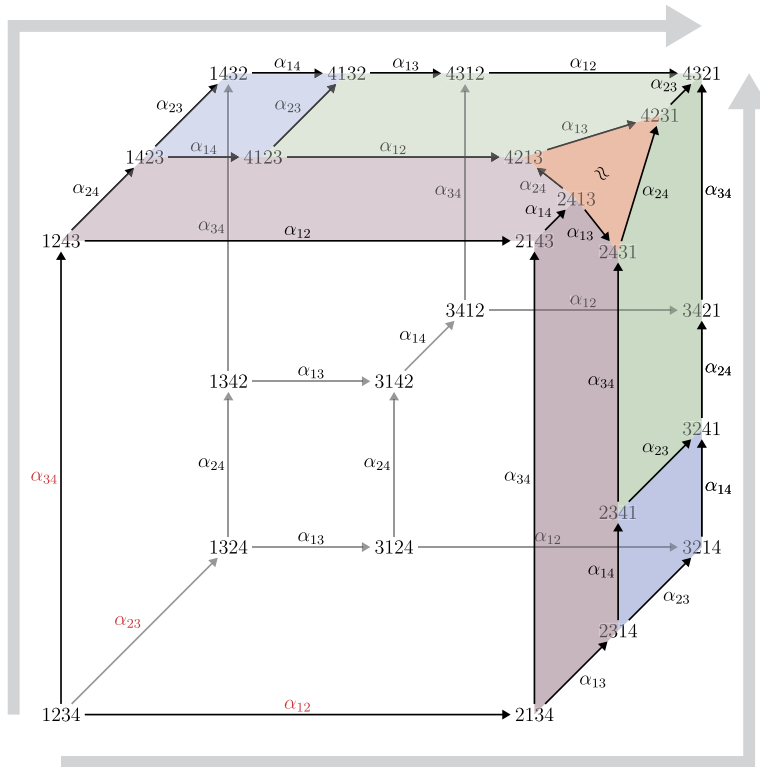
Fig. 11. Discrepancy in the cube identity.

Remark 7.44. Note that Fig. 11 is a 'cubification' of the permutohedron P_4. Here an interesting breakdown of symmetry in the cube identity is displayed (not unexpected, since the three overlapping pairs in $1\,\overline{2}\,\overline{3}\,4$ contain a middle one). As we have proved, the cube identity is valid up to convertibility. But it almost holds for 4 strands ($n = 4$) literally. If $A = \alpha_{12}$, $B = \alpha_{23}$ and $C = \alpha_{34}$, then

$$(B/C)/(A/C) \quad \approx (\neq) \quad (B/A)/(C/A)$$

The difference is one application of $\alpha_{13}\alpha_{24} = \alpha_{24}\alpha_{13}$ as in the red square in Fig. 11, or in Artin's notation $\mathbb{1}\mathbb{3} = \mathbb{3}\mathbb{1}$. But

$$(C/A)/(B/A) \quad = \quad (C/B)/(A/B)$$
$$(A/B)/(C/B) \quad = \quad (A/C)/(B/C)$$

Proposition 7.45 (Left and right cancellation).

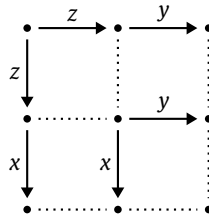(i) *For positive braid codes we have left cancellation:*

$$zx \approx zy \implies x \approx y$$

(ii) *For positive braid codes we also have right cancellation:*

$$xz \approx yz \implies x \approx y$$

Proof. (i) From $zx \approx zy$ follows $zx/zy = zy/zx = \varepsilon$ by Theorem 7.39. Because $z/z = \varepsilon$ and $x/\varepsilon = x$ and $y/\varepsilon = y$, we have $x/y = y/x = \varepsilon$:

Therefore $x \approx y$, by Theorem 7.39.

(ii) First observe that $x \approx y \Rightarrow \mathsf{inv}(x) \approx \mathsf{inv}(y)$. Suppose $xz \approx yz$. Then $\mathsf{inv}(xz) \approx \mathsf{inv}(yz)$, hence $\mathsf{inv}(z)\mathsf{inv}(x) \approx \mathsf{inv}(z)\mathsf{inv}(y)$. Now by left cancellation: $\mathsf{inv}(x) \approx \mathsf{inv}(y)$. Hence $x = \mathsf{inv}(\mathsf{inv}(x)) \approx \mathsf{inv}(\mathsf{inv}(y)) = y$. □

8. Concluding remarks and questions

At the end of this paper some remarks in hindsight may be in order. First: In Section 7 we went to considerable length to establish e.g. confluence-by-tiling, or the Cube Equation. It should be stressed that we did not aim for the shortest proof of these properties, but rather have tried to present a sample of relevant techniques from term rewriting to address this matter, such as the method of decreasing diagrams. Indeed, modern expositions can be found in Dehornoy [6,7] that establish these key theorems considered here much faster and much more general. As to the vast generality of the current state of the theory, pertaining to much more monoids than just the braids monoid, also extending to categories, we refer to [36].

8.1. Term rewriting techniques applied in braid and Garside theory

Historical reflections   Completing reduction diagrams by tiling is a classic technique in lambda calculus and in particular in term rewriting. The use of elementary reduction diagrams with empty steps was developed in [8].[4] The notion of the projection operator is prominent in early work of Lévy [5] and Huet [2–4]. In particular Lévy's diagram equations and the Cube Equation play an important role in that work.

The objective in those developments in lambda calculus and (orthogonal) term rewriting theory was confluence of reductions, a single reduction relation or an indexed family of rewrite relations. In braid theory there is also a major interest in completed reduction diagrams in order to obtain confluence.[5] We note that the terminology in that field has developed in an independent but quite compatible way. The terminology is left and right word reversing with an important concern for termination of left/right word reversing.[6] Also the projection operator (residual operator as Lévy called it), including the Cube Equation that we also needed and have derived in the present paper, plays an important role in braid theory and Garside Theory. [7]

Future prospects   We are hopeful that there is not only a clearly visible historical common ground in the technical sense between these two areas, lambda calculus and term rewriting (including string rewriting) versus braids and Garside theory, but we are also hopeful that there may be a future cross-fertilisation by mutual inspiration. From the side of term/string rewriting there is a large tool set that may turn out to be relevant for Garside theory. Such potentially useful tools include:

(i) decreasing diagrams (part of abstract rewriting theory)
(ii) residual theory,
(iii) termination methods,
(iv) critical pair completion,
(v) axiomatic rewrite theory,
(vi) string rewrite systems and Tietze moves.

8.2. Complete rewriting systems for the braid monoid

In this paper we have not considered, for reasons of space, the interesting theme of complete rewrite systems for positive braids, except for a sketch of Garside's normal form theorem. There are several versions of complete rewrite systems yielding normal forms. Krammer [37] contains some elegant and simple complete rewrite systems, for the Artin notation. For general methods in a group-theoretic context to obtain complete rewrite systems for Artin monoids and others, see [38]. There

---

[4]   An example of proofs exploiting empty steps in reduction diagrams is also in [7, page 219, Figure 11].

[5]   Indeed, it is mentioned by Dehornoy et al. [7, page 89], that the word reversing method is in particular useful for 'complemented presentations', proving cancellativity and existence of least common multiples.

[6]   See work of Dehornoy et al. [6, Chapter II, Word reversing] and [7, pages 68–71].

[7]   Lévy's diagram projection equations are used in [7, page 43, Corollary 2.13]. See also [7, page 76, Proposition 4.34, and page 154, Notation 3.38].

are also complete rewrite systems obtained by critical pair completion, see e.g. Bangert [22]. In fact, there are several different notions of normal forms for braids, and there exist very efficient algorithms to compute them. See for instance the 'relaxation algorithms' of [39].

(i) We expect that analogous versions of such complete rewrite systems as the ones in Krammer [37] can be given for braid codes.

Here an interesting question turns up, in view of the Kapur–Narendran phenomenon for the monoid for braids on 3 strands

$$\langle\, a, b \mid aba = bab \,\rangle$$

While the equality is clearly decidable, as all convertible words have the same length, there is no finite complete rewrite system yielding this equality, at least not in the same signature. However, an extension with a new defined constant, obtained via a Tietze move, with subsequent critical pair completion, does give a complete rewrite system. A question signalled also elsewhere [40], is whether this Kapur–Narendran phenomenon also pertains to braids with more than 3 strands.

(ii) Another question emerging from the focus in the present paper, is whether possible versions of complete rewrite systems in the braid code also 'suffer' from the Kapur–Narendran observation, not having a complete rewrite system in spite of their decidable convertibility.

(iii) At a deeper level, beyond the scope of this paper, one may wonder for the braid codes about the notion of 'finite derivation type' in the sense of Squier [41], referring to deep properties of monoid presentations whose absence of finite derivation type forbid the existence of complete rewrite systems for certain decidable monoid presentations, in a strong sense, namely even for all possible extensions with new constants added via Tietze moves.

Acknowledgements

References

[1] S. Abramsky, Temperley–Lieb algebra: from knot theory to logic and computation via quantum mechanics, arXiv preprint, arXiv:0910.2737.

[2] G. Huet, Confluent reductions: abstract properties and applications to term rewriting systems: abstract properties and applications to term rewriting systems, J. ACM 27 (4) (1980) 797–821.

[3] G. Huet, Residual theory in λ-calculus: a formal development, J. Funct. Programming 4 (3) (1994) 371–394.

[4] G.P. Huet, J.-J. Lévy, Computations in orthogonal rewriting systems, ii, in: Computational Logic-Essays in Honor of Alan Robinson, 1991, pp. 415–443.

[5] J.-J. Lévy, An algebraic interpretation of the λβk-calculus; and an application of a labelled λ-calculus, Theoret. Comput. Sci. 2 (1) (1976) 97–114.

[6] P. Dehornoy, Braids and Self-Distributivity, vol. 192, Birkhäuser, 2012.

[7] P. Dehornoy, F. Digne, E. Godelle, D. Krammer, J. Michel, Foundations of Garside Theory, Tracts in Mathematics, vol. 22, European Mathematical Society, 2015.

[8] J.W. Klop, Combinatory Reduction Systems, Mathematical Centre Tracts, vol. 127, Mathematisch Centrum, 1980.

[9] J.W. Klop, Term rewriting systems, in: Handbook of Logic in Computer Science, vol. II, Oxford University Press, 1992, pp. 1–116.

[10] Terese, Term Rewriting Systems, Cambridge Tracts in Theoretical Computer Science, vol. 55, Cambridge University Press, 2003.

[11] P.-A. Mellies, Axiomatic rewriting theory vi: residual theory revisited, in: Rewriting Techniques and Applications, Springer, 2002, pp. 5–11.

[12] J.H. Przytycki, Classical roots of knot theory, Chaos Solitons Fractals 9 (4–5) (1998) 531–545.

[13] E. Artin, Theorie der Zöpfe, Abh. Math. Semin. Univ. Hambg. 4 (1) (1925) 47–72.

[14] E. Artin, Braids and permutations, Ann. of Math. (1947) 643–649.

[15] E. Artin, Theory of braids, Ann. of Math. (1947) 101–126.

[16] F. Garside, The Theory of Knots and Associated Problems, Ph.D. thesis, University of Oxford, 1965.

[17] F.A. Garside, The braid group and other groups, Q. J. Math. 20 (1) (1969) 235–254.

[18] E. Brieskorn, K. Saito, Artin-gruppen und coxeter-gruppen, Invent. Math. 17 (1972) 245–271.

[19] J.S. Birman, T.E. Brendle, Braids: a survey, in: Handbook of Knot Theory, 2005, pp. 19–103.

[20] C. Kassel, V. Turaev, Braid Groups, Graduate Texts in Mathematics, vol. 247, 2008.

[21] M. Gardner, New Mathematical Diversions, Mathematical Association of America, 1995.

[22] P.D. Bangert, Braids and knots, in: Lectures on Topological Fluid Mechanics, Springer, 2009, pp. 1–73.

[23] G. Schmidt, T. Ströhlein, Relations and Graphs: Discrete Mathematics for Computer Scientists, Springer Science & Business Media, 2012.

[24] M. Chiodo, An Introduction to Braid Theory, Msc, University of Melbourne, 2005.

[25] P. Dehornoy, Groups with a complemented presentation, J. Pure Appl. Algebra 116 (1) (1997) 115–137.

[26] D.B.A. Epstein, M.S. Paterson, J.W. Cannon, D.F. Holt, S.V. Levy, W.P. Thurston, Word Processing in Groups, A. K. Peters, Ltd., 1992.

[27] J. Endrullis, D. Hendriks, J.W. Klop, Degrees of streams, J. Integers B 11 (2011) 1–40.

[28] J. Endrullis, J. Klop, A. Saarela, M. Whiteland, Degrees of transducibility, in: Proc. Conf. on Combinatorics on Words, WORDS 2015, in: LNCS, vol. 9304, Springer, 2015, pp. 1–13.

[29] J. Endrullis, J. Karhumäki, J. Klop, A. Saarela, Degrees of infinite words, polynomials and atoms, in: Proc. Conf. Developments in Language Theory, DLT 2016, in: LNCS, Springer, 2016, pp. 164–176.

[30] I. Bethke, J.W. Klop, R. de Vrijer, Descendants and origins in term rewriting, Inform. and Comput. 159 (1–2) (2000) 59–124.

[31] M. Bezem, J.W. Klop, V. Van Oostrom, Diagram techniques for confluence, Inform. and Comput. 141 (2) (1998) 172–204.

[32] N.G. de Bruijn, A Note on Weak Diamond Properties, Technische Hogeschool Eindhoven, Department of Mathematics, 1978.

[33] J. Endrullis, J.W. Klop, De Bruijn's weak diamond property revisited, Indag. Math. 24 (4) (2013) 1050–1072.
[34] V. Van Oostrom, Confluence by decreasing diagrams, Theoret. Comput. Sci. 126 (2) (1994) 259–280.
[35] J. Endrullis, J.W. Klop, R. Overbeek, Decreasing diagrams with two labels are complete for confluence of countable systems, in: Proc. Conf. on Formal Structures for Computation and Deduction, FSCD 2018, in: LIPIcs, vol. 108, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2018, pp. 14:1–14:15.
[36] P. Dehornoy, A cancellativity criterion for presented monoids, arXiv preprint, arXiv:1802.04607.
[37] D. Krammer, MA4F2 Braids Groups 2005–2006, Lecture Notes, 2006.
[38] S.M. Hermiller, J. Meier, Artin groups, rewriting systems and three-manifolds, J. Pure Appl. Algebra 136 (2) (1999) 141–156.
[39] P. Dehornoy, I. Dynnikov, D. Rolfsen, B. Wiest, Ordering Braids, Mathematical Surveys and Monographs, vol. 148, American Mathematical Society, 2008.
[40] Y. Guiraud, P. Malbos, S. Mimram, A homotopical completion procedure with applications to coherence of monoids, in: RTA-24th International Conference on Rewriting Techniques and Applications-2013, vol. 21, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2013, pp. 223–238.
[41] C.C. Squier, F. Otto, Y. Kobayashi, A finiteness condition for rewriting systems, Theoret. Comput. Sci. 131 (2) (1994) 271–294.
[42] P.-A. Melliès, Braids described as an orthogonal rewriting system, festschrift in Honour of Roel de Vrijer, 2009.
[43] G. Gonthier, J.-J. Lévy, P.-A. Mellies, An abstract standardisation theorem, in: Proc. Symp. on Logic in Computer Science (LICS), IEEE, 1992, pp. 72–81.
[44] J.W. Klop, V. van Oostrom, R. de Vrijer, Course notes on braids, Available on the authors' web page.