A Geometrical Procedure for Computing Relaxation ⋆

Gabriele Pulcini

Département d'Informatique de l'École Normale Supérieure
45, rue d'Ulm – F-75230 Paris Cedex 05 – France
pulcini@di.ens.fr

Abstract. Permutative logic is a non-commutative conservative extension of linear logic suggested by some investigations on the topology of linear proofs. In order to syntactically reflect the fundamental topological structure of orientable surfaces with boundary, permutative sequents turn out to be shaped like *q-permutations*. *Relaxation* is the relation induced on q-permutations by the two structural rules *divide* and *merge*; a decision procedure for relaxation has been already provided by stressing some standard achievements in theory of permutations. In these pages, we provide a parallel procedure in which the problem at issue is approached from the point of view afforded by geometry of 2-manifolds and solved by making specific surfaces interact.

Keywords: non-commutative linear logic, permutative logic.

1 Introduction

Permutative logic (PL) [3, 11] is a non-commutative variant of linear logic [7] suggested by some topological investigations on the geometry of linear proofs [4, 10, 6]. In order to syntactically reflect the basic topological information concerning orientable surfaces with boundary, permutative sequents turn out to be shaped like *q-permutations*: very simple combinatorial structures essentially consisting in a permutation σ indexed with a non-negative integer q [3, 12]. The structure of q-permutations is rooted in the well-known statement of the classification theorem which ensures that any orientable surface (possibly with boundary) is always homeomorphic either to a sphere or to a connected sum of tori (possibly with boundary) [8]. Roughly speaking, in a q-permutation σ_q, σ denotes, cycle by cycle, each boundary component, whereas the index q works as a counter for the number of tori involved in the connected sum to which the surface at issue is homeomorphic ($q = 0$ in case of a sphere).

PL is a non-commutative deductive system in the sense that some non-trivial exchanges can be performed only by letting the topology of the surfaces expressed by sequents evolve. At syntactical level, this specific kind of topological evolution is taken into account by two structural rules, *divide* and *merge*, and *relaxation* is the binary relation induced by these transformations on the set of q-permutations. Unlike other non-commutative variants of linear logic (*cyclic logic* [13], *non-commutative logic* [2], *planar logic* [9]) – for which the decision of relaxation is just a trivial question – the problem of checking whether two q-permutations are in relation of relaxation constitutes an interesting mathematical problem. A decision procedure for relaxation has been already provided by stressing some standard results in the theory of permutations [3].

The original contribution here proposed is rooted on a previous work devoted to q-permutations and derived structures as instruments for handling with surfaces and polygonal presentations [12]. In particular – as far as orientable surfaces are concerned – q-permutations have been shown able to induce a very effective algorithm for computing the quotient surface associates with any polygonal presentation. By stressing this achievement, we introduce a parallel procedure in which the problem of relaxation is approached from the point of view afforded by geometry of 2-manifolds and solved by making surfaces interact.

This specific contribution should be seen in the line of a wider project centred on a dialogue between geometry and logical structures which has been inaugurated by the already-mentioned investigations on the topology of linear proofs and more recently further developed through the works on Permutative Logic [3, 11] and combinatorial approaches to geometry of surfaces [12].

⋆ Research supported by the *Mairie de Paris*.

2 Orientable surfaces and q-Permutations

2.1 Surfaces and Polygonal Presentations

We deal with compact and connected 2-dimensional manifolds [8]; in the sequel of this paper, we will simply call them *surfaces*. It is a well-known achievement in algebraic topology that, thanks to the triangularisation theorem, any surface $\mathscr{S}$ can be univocally determined by a polygon $W_{\mathscr{S}}$ (which is not unique, but depending on the specific triangularisation performed on $\mathscr{S}$); in particular, $W_{\mathscr{S}}$ is such that:

- the perimeter of $W_{\mathscr{S}}$ is constituted by labelled and oriented edges;
- no more than two edges can have the same label;
- the quotient surface associated with $W_{\mathscr{S}}$ and obtained through identification of paired edges, is the just $\mathscr{S}$ (considered up to homeomorphisms) [8] .

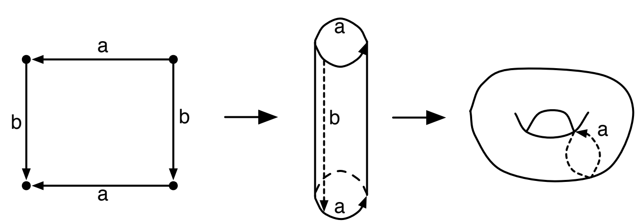
Since given a clockwise or an anticlockwise orientation, any polygon W turns out to be completely determined by its perimeter, namely by a cycle of oriented edges. Edges having orientation opposite to the fixed one are usually indicated by raising them at the minus one power. Therefore, a polygon is usually written as of word on an alphabet $\mathcal{A} \cup \mathcal{A}^{-1}$, where $\mathcal{A} = \{a, b, c, \ldots\}$ and $\mathcal{A}^{-1} = \{a^{-1}, b^{-1}, c^{-1}, \ldots\}$, considered up to circular permutations. In the sequel of this paper we will adopt the simplified notation x and $\bar{x}$ ($x \in \mathcal{A}$), for a pair of edges labeled with x having opposite orientations; the bar-operation ($\bar{\ }$) is clearly an involution without fixed point, namely, for any $x \in \mathcal{A}$, $\bar{\bar{x}} = x$ and $x \neq \bar{x}$.

The well-known classification theorem establishes that any surface (possibly with boundary) turns out to be homeomorphic to exactly one of the following surfaces: a sphere, a finite connected sum of tori, or a finite connected sum of projective planes (possibly with boundary). Sphere and connected sums of tori are orientable surfaces, whereas connected sums of projective planes are non-orientable [8]. Below, we recall some basic results concerning polygonal presentations:

- If in $W_{\mathscr{S}}$ occurs a letter z which is not paired, then $\mathscr{S}$ is bordered (z constitutes a "residual" edge which has to occur on the boundary).
- If in $W_{\mathscr{S}}$ all paired edges have opposite orientations, then $\mathscr{S}$ is orientable; otherwise it is non-orientable.

In this work we will limited our attention to orientable surfaces. We just recall two basic polygonal configurations concerning this kind of surfaces: *sphere* $\boldsymbol{a\bar{a}}$, *torus* $\boldsymbol{ab\bar{a}\bar{b}}$ (see Figure 1).

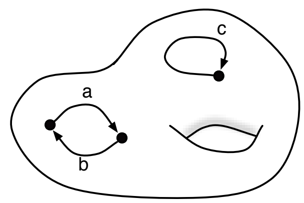
Fig. 1. Forming a torus.



2.2 q-Permutations

Let $\mathscr{S}$ be the quotient surface associated with a polygon $W_{\mathscr{S}}$; we respectively denote with $\partial W_{\mathscr{S}}$ the set of non-paired edges occurring on the perimeter of $W_{\mathscr{S}}$. It has already been remarked that: if $\partial W_{\mathscr{S}} \neq \varnothing$, then $\mathscr{S}$ is a bordered surface and, moreover: (i) all and only the edges in $\partial W_{\mathscr{S}}$ appear on the boundary, (ii) each

Fig. 2. Torus with boundary decomposed into two components.



boundary component is formed by at least one edge. We denote with $\partial\mathscr{S}$ the set of edges occurring on the boundary of $\mathscr{S}$; since given an orientation, we can notice that $\mathscr{S}$ induces a cyclic order on each of the disjoint subsets of $\partial\mathscr{S}$ corresponding to boundary-components: in this way, we obtain a permutation σ on $\partial\mathscr{S}$. The idea leading to the notion of q-permutation is that the basic information concerning any orientable surface $\mathscr{S}$ can always be encoded by a very easy mathematical structure consisting in a permutation σ (denoting, cycle by cycle, each boundary-component) indexed with a $q \in \mathbb{N}$ (used for counting tori in the connected sum to which $\mathscr{S}$ is homeomorphic). We report below a more formal definition for q-permutations.

Definition 1 (q-permutation). *A q-permutation α is an ordered triple (X, σ, q), where X is a finite support in which any letter may occur at most twice and paired edges have always opposite orientations, σ is a permutation on X and q a non-negative integer. Limit cases of q-permutations are those ones of the shape $(\varnothing, \epsilon, q)$, where ϵ denotes the permutation having empty domain.*

Notation. q-Permutations are indicated with small Greek letters $\alpha, \beta, \gamma, \ldots$. Big Greek letters $\Gamma, \Delta, \Lambda, \ldots$ and $\Sigma, \Xi, \ldots$ respectively denote finite lists of elements ($\Gamma = a_1, a_2, \ldots, a_n$) and finite sets of lists $\Sigma = \{(\Gamma_1), (\Gamma_2), \ldots, (\Gamma_n)\}$. As usual in algebra, when a list is included between brackets, it is intended to form a cycle and permutations are represented as products of disjoint cycles. When the specification of the domain is superfluous, q-permutations will be simply written as indexed permutations $\alpha = \{(\Gamma_1), (\Gamma_2), \ldots, (\Gamma_n)\}_q$. Moreover: if $\Gamma = a_1, a_2, \ldots, a_n$ and $\Sigma = \{(\Gamma_1), (\Gamma_2), \ldots, (\Gamma_n)\}$, then $\bar{\Gamma} = \bar{a}_n, \bar{a}_{n-1}, \ldots, \bar{a}_1$ and $\bar{\Sigma} = \bar{\Gamma}_1, \bar{\Gamma}_2, \ldots, \bar{\Gamma}_n$.

In [3, 12], q-permutations have been shown able to constitute a structure alternative to the classical one based on words; in fact, q-permutations, like words, are able to encode the basic topological information concerning orientable surfaces possibly with boundary (where each boundary-component is shaped like a polygon). In particular, an oriented surface $\mathscr{S}$ homeomorphic to the connected sum of q tori and having the boundary decomposed into k components $\Gamma_1, \Gamma_2, \ldots, \Gamma_k$ induces the q-permutation $\alpha_{\mathscr{S}} = \{(\Gamma_1), (\Gamma_2), \ldots, (\Gamma_k)\}_q$; vice versa a q-permutation $\alpha = \{(\Gamma_1), (\Gamma_2), \ldots, (\Gamma_k)\}_q$ denotes an oriented surface $\mathscr{S}_\alpha$ homeomorphic to the connected sum of q tori and having the boundary decomposed into k components $\Gamma_1, \Gamma_2, \ldots, \Gamma_k$.

Example 1. The surface proposed in Figure 2 is homeomorphic to a torus and its boundary is decomposed into two components: "(a, b)" and "(c)". Therefore, its corresponding q-permutation is $\{(a, b), (c)\}_1$.

We remark that q-permutations have to be considered modulo the transformation $\{\Sigma\}_q \sim \{\bar{\Sigma}\}_q$ which expresses the fact that the orientation according to which we decide to "read" the boundary of $\mathscr{S}$ is absolutely arbitrary.

2.3 Computing Surfaces via q-Permutations

q-Permutations have been recently shown able to induce a very easy and natural algorithm for computing the quotient surface $\mathscr{S}$ associated with any given polygonal presentation $W_{\mathscr{S}}$ [12]. As far as orientable surfaces are concerned, the reader can find the procedure at issue summarised below.

Definition 2 (system $\mathcal{P}$). *The rewriting system $\mathcal{P}$ is formed by q-permutations (terms) and by the following two rewriting rules:*

- cylinder: $\{\Sigma, (\Gamma, a, \Delta, \bar{a})\}_q \rightarrow \{\Sigma, (\Gamma), (\Delta)\}_q$
- torus: $\{\Sigma, (\Gamma, a), (\bar{a}, \Delta)\}_q \rightarrow \{\Sigma, (\Gamma, \Delta)\}_{q+1}$.

In [12], $\mathcal{P}$ has been shown able to enjoy both the strong normalisation property and the uniqueness of the normal form.

Definition 3. *By stressing the fact that a polygon is homeomorphic to a sphere with connected boundary, we can associate with any polygon $W_{\mathscr{S}}$ a q-permutation $\alpha_{W_{\mathscr{S}}}$ in the following way:*

$$\text{if } W_{\mathscr{S}} = a_1 a_2 \dots a_n, \text{ then } \alpha_{W_{\mathscr{S}}} = \{(a_1, a_2, \dots, a_n)\}_0.$$

Theorem 1. *Let $W_{\mathscr{S}}$ be a polygonal presentation associated with the surface $\mathscr{S}$ and $\alpha'_{W_{\mathscr{S}}}$ the normal form of $\alpha_{W_{\mathscr{S}}}$ (in symbols: $\alpha_{W_{\mathscr{S}}} \rightsquigarrow^*_{\mathcal{P}} \alpha'_{W_{\mathscr{S}}}$). $\alpha'_{W_{\mathscr{S}}}$ exactly denotes $\mathscr{S}$.*

Proof. Essentially obtained by remarking that the system $\mathcal{P}$ works by faithfully following, step by step, the process of forming a surface through identification of paired edges. The reader can find all the details in [12].

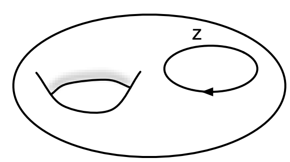
This theorem provides a very easy and effective procedure for computing the quotient surface associated with any given polygonal presentation; some examples are afforded below.

Example 2. The process of forming a torus illustrated in Figure 1 can be reproduced in terms of q-permutations as follows:

$$\{(\bar{b}, \bar{a}, b, a)\}_0 \rightarrow_{\text{cyl.}} \{(\bar{a}), (a)\}_0 \rightarrow_{\text{torus}} \epsilon_1.$$

Example 3. Wa aim to compute the surface presented by the polygon $a\bar{b}\bar{c}zb\bar{a}c$. According to Theorem 1, we rewrite the q-permutation $\{(a, \bar{b}, \bar{c}, z, b, \bar{a}, c)\}_0$ as indicated below. The resulting surface is that one characterized by $\{(z)\}_1$, i.e. a torus with an unique boundary-component "(z)".

$$\begin{aligned} &\{(a, \bar{b}, \bar{c}, z, b, \bar{a}, c)\}_0 \rightarrow_{\text{cyl.}} \\ &\rightarrow_{\text{cyl.}} \{(\bar{b}, \bar{c}, z, b), (c)\}_0 \rightarrow_{\text{torus}} \\ &\rightarrow_{\text{torus}} \{(\bar{b}, z, b)\}_1 \rightarrow_{\text{cyl.}} \{(z)\}_1 \end{aligned}$$



3 Relaxation in Permutative Logic

In order to syntactically reflect the fundamental structure of orientable surfaces with boundary, PL sequents turn out to be shaped like q-permutations. In PL, the surface expressed by q-permutations can be evolved through two structural rules: *divide* and *merge*.

Definition 4. *$\mathcal{S}$ is the rewriting system on q-permutations formed by the four following structural rules:*

$$\frac{\{\Sigma, (\Gamma, \Delta)\}_q}{\{\Sigma, (\Gamma), (\Delta)\}_q}\ \textit{divide}, \qquad \frac{\{\Sigma, (\Gamma)\}_q}{\{\Sigma, (\Gamma), ()\}_q}\ \textit{empty divide},$$

$$\frac{\{\Sigma, (\Gamma), (\Delta)\}_q}{\{\Sigma, (\Gamma, \Delta)\}_{q+1}}\ \textit{merge}, \qquad \frac{\{\Sigma, (\Gamma), ()\}_q}{\{\Sigma, (\Gamma)\}_{q+1}}\ \textit{empty merge}.$$

On the one hand, the divide operation consists in evolving the topology of a bordered surface by identifying two new opposite edges, a and $\bar{a}$, opened on the same boundary component (see Figure 3). On the other hand, in case of a merge-evolution, new opposite edges are opened on two different boundary-components and so their identification produces a new handle on the surface, i.e. one more torus in the connected sum (see Figure 3).

$$\frac{\{\Sigma,(\Gamma,\Delta)\}_q}{\{\Sigma,(\Gamma),(\Delta)\}_q}\ \text{divide} \quad\cong\quad \frac{\dfrac{\{\Sigma,(\Gamma,\Delta)\}_q}{\{\Sigma,(\Gamma,a,\Delta,\bar{a})\}_q}\ \text{insert new edges}}{\{\Sigma,(\Gamma),(\Delta)\}_q}\ \text{cylinder}$$

$$\frac{\{\Sigma,(\Gamma),(\Delta)\}_q}{\{\Sigma,(\Gamma,\Delta)\}_{q+1}}\ \text{merge} \quad\cong\quad \frac{\dfrac{\{\Sigma,(\Gamma),(\Delta)\}_q}{\{\Sigma,(\Gamma,a),(\bar{a},\Delta)\}_q}\ \text{insert new edges}}{\{\Sigma,(\Gamma,\Delta)\}_{q+1}}\ \text{torus}$$

Fig. 3. Evolution through a divide rule.

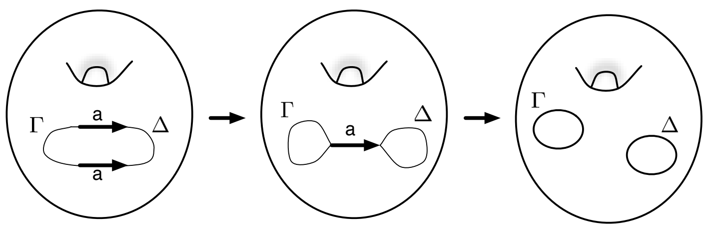
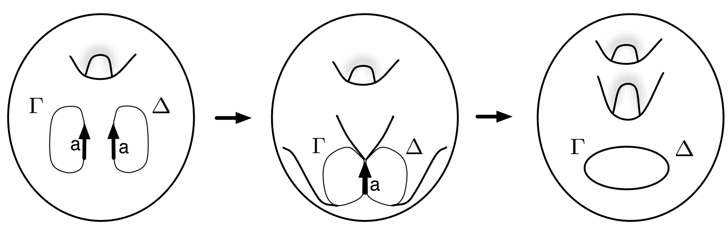


Fig. 4. Evolution through a torus rule.



Definition 5. *Given two q-permutations α and β, we say that α* relaxes to *β ($\alpha \succeq \beta$) if, and only if, there exists a chain $\mathscr{C} : \alpha \rightsquigarrow_{\mathcal{S}} \beta$.*

Example 4. It is easy to check that whereas $\{(a,b),(c)\}_1 \succeq \{(a,c),(b)\}_2$, the q-permutation $\{(a,c),(b)\}_1$ cannot relax to $\{(a,b),(c)\}_1$.

Because of the fact that each single application of divide or merge on a q-permutation α increases the rank of the denoted surface $\mathscr{S}_\alpha$, we have that relaxation induces a partial order on the set of q-permutations [3].

The following definition is based on the obvious remark that, by removing the superfluous information concerning indices, $\mathcal{S}$ can be seen as a rewriting system working directly on permutations.

Definition 6. *The minimum number of applications of the merge rule needed for rewriting a permutation σ into another permutation τ ($\sigma, \tau \in S_n$), is called the* distance *between σ and τ and denoted with $d(\sigma,\tau)$.*

Remark 1. – d is a total function, namely, given any couple of permutations $\sigma, \tau \in S_n$, there always exists a chain carrying σ into τ: it is sufficient to consider the limit case in which a series of divide applications detaches each single element of σ into a cycle, then τ can be constructed element by element through a final series of merge rules.
– Although $d(\sigma, \tau)$ is a nonsymmetric function, it enjoys the triangle inequality and therefore it deserves to be called a distance.

Theorem 2 (computing the distance). *Let $\sigma, \tau \in S_n$ be two permutations. We have:*

$$d(\sigma, \tau) = \frac{n - (\sigma^{-1}\tau)^{\bullet} + \sigma^{\bullet} - \tau^{\bullet}}{2},$$

where $\sigma^{\bullet}$ and $\tau^{\bullet}$ respectively denote the number of cycles of σ and τ.

Proposition 1. *Let $\alpha = \sigma_d$ and $\beta = \tau_e$ two q-permutations: $\alpha \succeq \beta$ if, and only if, $d(\sigma, \tau) \leqslant e - d$.*

The decision procedure provided in [3] is obtained by combining Proposition 1 and Theorem 2. Below we propose a concrete application of this algorithm.

Example 5. We draw on the two cases already analysed in Example 4. By applying Theorem 2, it is easy to check that for rewriting $\sigma = (a, b)(c)$ into $\tau = (a, c)(b)$ at least one application of the merge rule is needed, in other words, $d(\sigma, \tau) = 1$. Therefore, by Proposition 1, we have that $\{(a, b), (c)\}_1 \succeq \{(a, c), (b)\}_2$, but $\{(a, b), (c)\}_1 \not\succeq \{(a, c), (b)\}_1$.

4 A Geometrical Decision Procedure for Relaxation

4.1 An Intermediate Procedure

Notation. With $\mathcal{S}^{\star}$ we denote the system $\mathcal{S}$ fitted to permutations, namely $\mathcal{S}^{\star}$ is obtained from $\mathcal{S}$ just by forgetting indices. The length of a chain $\mathscr{C}$ of $\mathcal{S}^{\star}$ (i.e. the number of $\mathcal{S}^{\star}$-transformations occurring in $\mathscr{C}$) is indicated by $lh(\mathscr{C})$. $div(\mathscr{C})$ and $mer(\mathscr{C})$ respectively denote the number of divide and merge rules occurring in a chain $\mathscr{C}$; $div^*(\mathscr{C})$ and $mer^*(\mathscr{C})$ respectively denote the number of non-empty divide and merge applications in $\mathscr{C}$.

Proposition 2. *In any $\mathcal{S}^{\star}$-chain $\mathscr{C} : \sigma \rightsquigarrow \tau$, $div^*(\mathscr{C}) - mer^*(\mathscr{C}) = \tau^{\bullet} - \sigma^{\bullet}$.*

Lemma 1. *Consider two permutations $\sigma, \tau \in S_n$. A chain $\mathscr{C} : \sigma \rightsquigarrow_{\mathcal{S}^{\star}} \tau$ is such that $\mathrm{mer}(\mathscr{C}) = d(\sigma, \tau)$ if, and only if, $lh(\mathscr{C})$ is minimal.*

Proof. Simply by Proposition 2 and by remarking that $lh(\mathscr{C}) = mer(\mathscr{C}) + div(\mathscr{C})$.

Thanks to the previous lemma, the problem of computing the distance between two permutations turns out to be equivalent to that one of producing a minimal $\mathcal{S}^{\star}$-chain from σ to τ.

Theorem 3. *Given two permutations $\sigma, \tau \in S_n$, any chain $\mathscr{C} : \sigma \rightsquigarrow_{\mathcal{S}^{\star}} \tau$ produced through arbitrary applications of the following two specific applications of divide and merge,*

$$\textit{if } \tau(a) = b : \frac{\Sigma, (\Gamma, a, \Delta, b)}{\Sigma, (\Gamma, a, b), (\Delta)} \textit{ divide}$$

$$\textit{if } \tau(a) = b : \frac{\Sigma, (\Gamma, a), (b, \Delta)}{\Sigma, (\Gamma, a, b, \Delta)} \textit{ merge,}$$

is such that $m(\mathscr{C}) = d(\sigma, \tau)$.

Example 6. In order to better understand the mechanism of the just provided procedure, we produce a chain from $\sigma = (a_0, a_1, a_3)(a_5, a_4, a_2, a_6)$ to $\tau = (a_0, a_4, a_3)(a_2, a_5)(a_1, a_6)$.

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{(a_0, a_1, a_3), (a_5, a_4, a_2, a_6)}{(a_0, a_4, a_2, a_6, a_5, a_1, a_3)}\text{ merge (in fact: } \tau(a_0) = a_4)}{(a_0, a_4, a_3), (a_2, a_6, a_5, a_1)}\text{ divide (in fact: } \tau(a_4) = a_3)}{(a_0, a_4, a_3), (a_2, a_5, a_1), (a_6)}\text{ divide (in fact: } \tau(a_2) = a_5)}{(a_0, a_4, a_3), (a_2, a_5), (a_1), (a_6)}\text{ divide (in fact: } \tau(a_5) = a_2)}{(a_0, a_4, a_3), (a_2, a_5), (a_1, a_6)}\text{ merge (in fact: } \tau(a_1) = a_6)$$

Proof. We remark that, for permutations, divide and merge are inverse of each other and their applications correspond to multiply by an appropriate transposition:

$$\Sigma(x, \Gamma, y, \Delta) \cdot (x, y) = \Sigma(x, \Gamma)(y, \Delta),$$

$$\Sigma(x, \Gamma)(y, \Delta) \cdot (x, y) = \Sigma(x, \Gamma, y, \Delta).$$

In this way, any chain $\mathscr{C} : \sigma \rightsquigarrow_{\mathcal{S}^\star} \tau$ constitutes a decomposition of $\sigma^{-1}\tau$ into a product of transpositions $(x_1, y_1)(x_2, y_2)\dots(x_{lh(\mathscr{C})}, y_{lh(\mathscr{C})})$. Hence, the problem of checking the minimality of $\mathscr{C}$ becomes the problem of checking whether $\mathscr{C}$ provides a minimal decomposition for $\sigma^{-1}\tau$. We stress a standard result in the theory of permutations which establishes that if a product of transpositions $(x_1, y_1)(x_2, y_2)\dots(x_n, y_n) = \rho$ is such that, for each i, y_i does not appear in all the successive transpositions $(x_{i+1}, y_{i+1})\dots(x_n, y_n)$, then it is a minimal decomposition for ρ [5]. In the specific case of the two rules reported in the statement of our theorem, the corresponding transpositions are in order (b, Δ_1), where Δ_1 is the first element of Δ, and (b, Γ_1), where Γ_1 is the first element of Γ: in both the cases we have the occurrences of a b which is the element we aim to put in the right place, just after a. So, it is clear that b cannot appear in the successive transpositions and by Lemma 1 we have the statement of our theorem.

4.2 Geometrical Interpretation

In this final paragraph, we provide a geometrical procedure for computing the distance between two given permutations $\sigma, \tau \in S_n$. The idea is that any permutation $\sigma = (\Gamma_1)(\Gamma_2)\dots(\Gamma_k)$ can be topologically conceived as an oriented disk (lower topological complexity) having the boundary decomposed into k components $(\Gamma_1), (\Gamma_2), \dots, (\Gamma_k)$. We show that the composition (as usual, through identification of paired edges) of the two disks topological counterparts of σ and τ, induces a quotient surface whose topological complexity provides the information needed for computing the distance $d(\sigma, \tau)$. Such a procedure is stated in Theorem 5 and proved by stressing Theorem 3. Theorem 3 says in fact that the procedure which consists in putting each single element in the right place produces geodesic trajectories from σ to τ, and the same mechanism can be recognised in the process of topological composition just mentioned.

Notation. $|\alpha|$ and $|\sigma|$ denote the support of, respectively, the q-permutation α and the permutation σ.

Definition 7. *We consider two q-permutations $\alpha = \{\Sigma, (\Gamma, z)\}_e$ and $\beta = \{\Xi, (\Delta, \bar{z})\}_f$ such that $|\alpha| = |\beta|$. The $*$-composition between α and β is obtained by "gluing" them along an arbitrary edge z of their supports: $\alpha *_z \beta = \{\Sigma, \Xi, (\Gamma, \Delta)\}_{e+f}$.*

Definition 8. *α_σ is the q-permutation associated with the permutation σ in the following way: if $\sigma = (\Gamma_1)(\Gamma_2)\dots(\Gamma_n)$, then $\alpha_\sigma = \{(\Gamma_1), (\Gamma_2), \dots, (\Gamma_n)\}_0$.*

Procedure 4 *We circumscribe a specific procedure for normalising q-permutations of the form $\alpha_\sigma *_z \alpha_\tau$. Suppose that $\alpha_\sigma = \{(\Gamma_1, z_1), (\Gamma_2), \dots, (\Gamma_{\sigma^\bullet})\}_e$ and $\alpha_\tau = \{(\Delta_1, z_1), (\Delta_2, z_2), \dots, (\Delta_{\tau^\bullet}, z_{\tau^\bullet})\}_f$. Together with the usual torus rule we stress the following decomposed versions of cylinder and torus:*

$$\frac{\{\Sigma,(\Gamma,a,\Delta,\bar a)\}_q}{\{\Sigma,(\Gamma),(\Delta)\}_q}\,\text{cylinder}^* \quad\mapsto\quad \dfrac{\dfrac{\{\Sigma,(\Gamma,a,\Delta,\bar a)\}_q}{\{\Sigma,(\Gamma,a,\bar a),(\Delta)\}_q}\,\text{divide}}{\{\Sigma,(\Gamma),(\Delta)\}_q}\,\text{trivial cylinder},$$

$$\frac{\{\Sigma,(\Gamma,a),(\Delta,\bar a)\}_q}{\{\Sigma,(\Gamma,\Delta)\}_{q+1}}\,\text{torus}^* \quad\mapsto\quad \dfrac{\dfrac{\{\Sigma,(\Gamma,a),(\Delta,\bar a)\}_q}{\{\Sigma,(\Gamma,a,\bar a,\Delta)\}_{q+1}}\,\text{merge}}{\{\Sigma,(\Gamma,\Delta)\}_{q+1}}\,\text{trivial cylinder}.$$

We normalise $\alpha_\sigma *_{z_1} \alpha_\tau = \{(\bar\Delta_{\tau\bullet},\bar z_{\tau\bullet}),\dots,(\bar\Delta_2,\bar z_2),(\bar\Delta_1,\Gamma_1),(\Gamma_2),\dots,(\Gamma_{\sigma\bullet})\}_0$ *as follows.*

1. *Suppose that* $\Delta_1 = a_1,a_2,\dots,a_r$. *At first, we "consume" the segment* $\bar\Delta_1$ *by identifying, in order, the edges labelled with* $a_1,a_2,\dots,a_n$*:*

$$\frac{\{(\bar\Delta_{\tau\bullet},\bar z_{\tau\bullet}),\dots,(\bar\Delta_2,\bar z_2),(\bar a_r,\bar a_{r-1},\dots,\bar a_1,\Gamma_1),(\Gamma_2),\dots,(\Gamma_{\sigma\bullet})\}_0}{\{(\bar\Delta_{\tau\bullet},\bar z_{\tau\bullet}),\dots,(\bar\Delta_2,\bar z_2),(\Gamma'_1),(\Gamma'_2),\dots,(\Gamma'_k)\}_q}\,\text{cylinder}^*/\text{torus}^*.$$

2. *Now suppose that* $\Gamma'_1 = \Gamma'', z_2$. *We "activate" the second cycle* $(\bar\Delta_2,\bar z_2)$ *of* τ *by identifying the* z_2*-edges through an usual torus rule:*

$$\dfrac{\dfrac{\{(\bar\Delta_{\tau\bullet},\bar z_{\tau\bullet}),\dots,(\bar\Delta_2,\bar z_2),(\bar a_r,\bar a_{r-1},\dots,\bar a_1,\Gamma_1),(\Gamma_2),\dots,(\Gamma_{\sigma\bullet})\}_0}{\{(\bar\Delta_{\tau\bullet},\bar z_{\tau\bullet}),\dots,(\bar\Delta_2,\bar z_2),(\Gamma'',z_2),(\Gamma'_2),\dots,(\Gamma'_k)\}_q}\,\text{cylinder}^*/\text{torus}^*}{\{(\bar\Delta_{\tau\bullet},\bar z_{\tau\bullet}),\dots,(\bar\Delta_2,\Gamma''),(\Gamma'_2),\dots,(\Gamma'_k)\}_{q+1}}\,\text{(activating) torus}.$$

3. *We iterate steps 1 and 2 until we reach the normal form.*

Example 7. According to Procedure 4 we normalise the q-permutation

$$\alpha_\sigma *_{a_0} \alpha_\tau = \{(\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,\bar a_4,a_1,a_3),(a_5,a_4,a_2,a_6)\}_0$$

coming from $\sigma = (a_0,a_1,a_3)(a_5,a_4,a_2,a_6)$ and $\tau = (a_0,a_4,a_3)(a_2,a_5)(a_1,a_6)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\{(\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,\bar a_4,a_1,a_3),(a_5,a_4,a_2,a_6)\}_0}{\{(\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,\bar a_4,a_4,a_2,a_6,a_5,a_1,a_3)\}_1}\,\text{merge}}{\{(\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,a_2,a_6,a_5,a_1,a_3)\}_1}\,\text{trivial cylinder}}{\{(\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,a_3),(a_2,a_6,a_5,a_1)\}_1}\,\text{divide}}{\{(\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(a_2,a_6,a_5,a_1)\}_1}\,\text{trivial cylinder}}{\{(\bar a_1,\bar a_6),(\bar a_5,a_6,a_5,a_1)\}_2}\,\text{(activating) torus}}{\{(\bar a_1,\bar a_6),(\bar a_5,a_5,a_1),(a_6)\}_2}\,\text{divide}}{\{(\bar a_1,\bar a_6),(a_1),(a_6)\}_2}\,\text{trivial cylinder}}{\{(\bar a_6),(a_6)\}_3}\,\text{(activating) torus}}{\{(\bar a_6,a_6)\}_4}\,\text{merge}}{\epsilon_4}\,\text{trivial cylinder}$$

Lemma 2. *If* $\mathscr{C}$ *is a chain afforded by Procedure 4, then in* $\mathscr{C}$ *torus and trivial cylinder applications permute downwards with both merge and divide applications.*

Proof. Some basic commutations are reported in Table 1.

$$\begin{array}{cl}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a},a,\varGamma',b,\varGamma''),\varXi\}_q & \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},\varGamma',b,\varGamma''),\varXi\}_q & \text{trivial cylinder} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},b,\varGamma''),(\varGamma'),\varXi\}_q & \text{divide} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\varGamma''),(\varGamma'),\varXi\}_q & \text{trivial cylinder}
\end{array}
\mapsto
\begin{array}{cl}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a},a,\varGamma',b,\varGamma''),\varXi\}_q & \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a},a,b,\varGamma''),(\varGamma'),\varXi\}_q & \text{divide} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},b,\varGamma''),(\varGamma'),\varXi\}_q & \text{trivial cylinder} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\varGamma''),(\varGamma'),\varXi\}_q & \text{trivial cylinder}
\end{array}$$

$$\begin{array}{cl}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a},a,\varGamma'),(b,\varGamma''),\varXi\}_q & \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},\varGamma'),(b,\varGamma''),\varXi\}_q & \text{trivial cylinder} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},b,\varGamma'',\varGamma'),\varXi\}_{q+1} & \text{merge} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\varGamma'',\varGamma'),\varXi\}_{q+1} & \text{trivial cylinder}
\end{array}
\mapsto
\begin{array}{cl}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a},a,\varGamma'),(b,\varGamma''),\varXi\}_q & \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a},a,b,\varGamma'',\varGamma'),\varXi\}_{q+1} & \text{merge} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},b,\varGamma'',\varGamma'),\varXi\}_{q+1} & \text{trivial cylinder} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\varGamma'',\varGamma'),\varXi\}_{q+1} & \text{trivial cylinder}
\end{array}$$

$$\begin{array}{cl}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a}),(a,\varGamma',b,\varGamma''),\varXi\}_q & \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},\varGamma',b,\varGamma''),\varXi\}_{q+1} & \text{(act.) torus} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},b,\varGamma''),(\varGamma'),\varXi\}_{q+1} & \text{divide}
\end{array}
\mapsto
\begin{array}{cl}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a}),(a,\varGamma',b,\varGamma''),\varXi\}_q & \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a}),(a,b,\varGamma''),(\varGamma'),\varXi\}_q & \text{divide} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},b,\varGamma''),(\varGamma'),\varXi\}_{q+1} & \text{(act.) torus}
\end{array}$$

$$\begin{array}{cl}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a}),(a,\varGamma'),(b,\varGamma''),\varXi\}_q & \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},\varGamma'),(b,\varGamma''),\varXi\}_{q+1} & \text{(act.) torus} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},b,\varGamma'',\varGamma'),\varXi\}_{q+2} & \text{merge}
\end{array}
\mapsto
\begin{array}{cl}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a}),(a,\varGamma'),(b,\varGamma''),\varXi\}_q & \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},\bar{a}),(a,b,\varGamma'',\varGamma'),\varXi\}_{q+1} & \text{merge} \\ \cline{1-1}
\{\varSigma,(\bar{\varDelta},\bar{b},b,\varGamma'',\varGamma'),\varXi\}_{q+2} & \text{(act.) torus}
\end{array}$$

Table 1. Basic commutations.

Example 8. We consider the chain reported in the previous example and we perform the permutations indicated by Lemma 2.

$$\begin{array}{cl}
\{(\bar{a}_1,\bar{a}_6),(\bar{a}_2,\bar{a}_5),(\bar{a}_3,\bar{a}_4,a_1,a_3),(a_5,a_4,a_2,a_6)\}_0 & \\ \cline{1-1}
\{(\bar{a}_1,\bar{a}_6),(\bar{a}_2,\bar{a}_5),(\bar{a}_3,\bar{a}_4,a_4,a_2,a_6,a_5,a_1,a_3)\}_1 & \text{merge} \\ \cline{1-1}
\{(\bar{a}_1,\bar{a}_6),(\bar{a}_2,\bar{a}_5),(\bar{a}_3,\bar{a}_4,a_4,a_3),(a_2,a_6,a_5,a_1)\}_1 & \text{divide} \\ \cline{1-1}
\{(\bar{a}_1,\bar{a}_6),(\bar{a}_2,\bar{a}_5),(\bar{a}_3,\bar{a}_4,a_4,a_3),(a_2,a_5,a_1),(a_6)\}_1 & \text{divide} \\ \cline{1-1}
\{(\bar{a}_1,\bar{a}_6),(\bar{a}_2,\bar{a}_5),(\bar{a}_3,\bar{a}_4,a_4,a_3),(a_2,a_5,a_1,a_6)\}_2 & \text{merge} \\ \cline{1-1} \cline{1-1}
\{(\bar{a}_1,\bar{a}_6),(\bar{a}_2,\bar{a}_5),(a_2,a_5,a_1,a_6)\}_2 & \text{trivial cylinder} \\ \cline{1-1}
\{(\bar{a}_1,\bar{a}_6),(\bar{a}_5,a_5,a_1,a_6)\}_3 & \text{(activating) torus} \\ \cline{1-1}
\{(\bar{a}_1,\bar{a}_6),(a_1,a_6)\}_3 & \text{trivial cylinder} \\ \cline{1-1}
\{(\bar{a}_6,a_6)\}_4 & \text{(activating) torus} \\ \cline{1-1}
\epsilon_4 & \text{trivial cylinder}
\end{array}$$

Theorem 5 (computing distance). *Let σ and τ be two permutations such that $|\sigma| = |\tau|$ and $z \in |\sigma|$. We have: $\alpha_\sigma *_z \alpha_\tau \rightsquigarrow^*_{\mathcal{P}} \epsilon_{d(\sigma,\tau)+\tau^\bullet-1}$.*

Proof. Let $\mathscr{C} : \alpha_\sigma *_z \alpha_\tau \rightsquigarrow \epsilon_q$ be a chain afforded by Procedure 4 and $\mathscr{C}'$ the chain obtained from $\mathscr{C}$ by performing the commutations indicated in Lemma 2. By remarking that the number of (activating) tori is always equal to $\tau^\bullet - 1$, $\mathscr{C}'$ will have the form:

$$\begin{array}{cl}
\{(\bar{\varDelta}_{\tau^\bullet}),\ldots,(\bar{\varDelta}_2),(\bar{\varDelta}_1,\varGamma_1),(\varGamma_2),\ldots,(\varGamma_{\sigma^\bullet})\}_0 & \\ \cline{1-1} \cline{1-1}
\{(\bar{\varDelta}_{\tau^\bullet}),\ldots,(\bar{\varDelta}_2),(\bar{\varDelta}_1,\varDelta_1),(\varDelta_2),\ldots,(\varDelta_{\tau^\bullet})\}_{q'} & \text{divide/merge} \\ \cline{1-1} \cline{1-1}
\epsilon_{q'+\tau^\bullet-1} & \text{trivial cylinder/(act.) torus.}
\end{array}$$

Now, we consider the first part of $\mathscr{C}'$: that one formed by divide and merge applications. By shrinking the segment $\left[\bar{\varDelta}_{\tau^\bullet}),\ldots,(\bar{\varDelta}_2),(\bar{\varDelta}_1\right]$ into the unique element z we obtain the $\mathcal{S}$-chain:

$$\frac{\{(z,\Gamma_1),(\Gamma_2),\dots,(\Gamma_{\sigma^\bullet})\}_0}{\{(z,\Delta_1),(\Delta_2),\dots,(\Delta_{\tau^\bullet})\}_{q'}}\ \text{divide/merge}$$

which can be seen – by the mechanism itself of Procedure 4 – as produced according to Theorem 3 (modulo some final divide applications) and so $q' = d(\sigma,\tau)$.

Example 9. As a final step we show that the chain obtained in Example 8 induces a minimal chain rewriting $\sigma = (a_0, a_1, a_3)(a_5, a_4, a_2, a_6)$ into $\tau = (a_0, a_4, a_3)(a_2, a_5)(a_1, a_6)$.

$$\begin{array}{l}
\dfrac{\{(\Big[\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,\bar a_4\Big],a_1,a_3),(a_5,a_4,a_2,a_6)\}_0}{\{(\Big[\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,\bar a_4\Big],a_4,a_2,a_6,a_5,a_1,a_3)\}_1}\ \text{merge} \\
\dfrac{}{\{(\Big[\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,\bar a_4\Big],a_4,a_3),(a_2,a_6,a_5,a_1)\}_1}\ \text{divide} \\
\dfrac{}{\{(\Big[\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,\bar a_4\Big],a_4,a_3),(a_2,a_5,a_1),(a_6)\}_1}\ \text{divide} \\
\dfrac{}{\{(\Big[\bar a_1,\bar a_6),(\bar a_2,\bar a_5),(\bar a_3,\bar a_4\Big],a_4,a_3),(a_2,a_5,a_1,a_6)\}_2}\ \text{merge}
\end{array} \mapsto$$

$$\mapsto \begin{array}{l}
\dfrac{\{(a_0,a_1,a_3),(a_5,a_4,a_2,a_6)\}_0}{\{(a_0,a_4,a_2,a_6,a_5,a_1,a_3)\}_1}\ \text{merge} \\
\dfrac{}{\{(a_0,a_4,a_3),(a_2,a_6,a_5,a_1)\}_1}\ \text{divide} \\
\dfrac{}{\{(a_0,a_4,a_3),(a_2,a_5,a_1),(a_6)\}_1}\ \text{divide} \\
\dfrac{}{\{(a_0,a_4,a_3),(a_2,a_5,a_1,a_6)\}_2}\ \text{merge} \\
\dfrac{}{\{(a_0,a_4,a_3),(a_2,a_5),(a_1,a_6)\}_2}\ \text{divide}
\end{array}$$

Figure 5 directly indicates the geometrical version of the procedure stated by Theorem 5. We consider the concrete case proposed in the previous example: the geometrical interpretations of τ and σ respectively consist in a disk having perimeter "$a_4a_3a_0$" and two boundary-components "a_2a_5" and "a_1a_6", and in a disk having perimeter "$a_0a_1a_3$" and one boundary-component "$a_5a_4a_2a_6$". In order to compute the distance between σ and τ, we compose their geometrical configurations obtaining an unique bordered polygon representing the connected sum of 4 tori. Hence, $\tau^\bullet = 3$ and so $d(\sigma,\tau) = 4 - 3 + 1 = 2$.

5 Future Work

In [12], the notion of q-permutation has been extended to that one of pq-permutation which allows to overcome the limit of orientability and so to characterise the whole range of topological surfaces, non-orientable included. Roughly speaking, pq-permutations are simply obtained from q-permutations by replacing the single index q with an ordered couple $\langle p,q\rangle$ of positive integers. Whereas the first index counts, as usual, the number of tori, the second one indicates projective planes. In order to deal with this kind of more general combinatorial structures, the rewriting system $\mathcal{P}$ turns out to be enriched with the following two "non-orientable" transformations:

$$\{\Sigma,(\Gamma,a,\Delta,a)\}_{\langle p,q\rangle} \rightsquigarrow_{\text{Mobius}} \{\Sigma,(\Gamma,\bar\Delta)\}_{\langle p,q+1\rangle}$$

and

$$\{\Sigma,(\Gamma,a),(a,\Delta)\}_{\langle p,q\rangle} \rightsquigarrow_{\text{Klein}} \{\Sigma,(\Gamma,\bar\Delta)\}_{\langle p,q+2\rangle},$$

which respectively induce the two following structural transformations:

$$\frac{\{\Sigma,(\Gamma,\Delta)\}_{\langle p,q\rangle}}{\{\Sigma,(\Gamma,\bar\Delta)\}_{\langle p,q+1\rangle}}\ \text{Möbius} \qquad \text{and} \qquad \frac{\{\Sigma,(\Gamma),(\Delta)\}_{\langle p,q\rangle}}{\{\Sigma,(\Gamma,\bar\Delta)\}_{\langle p,q+2\rangle}}\ \text{Klein.}$$

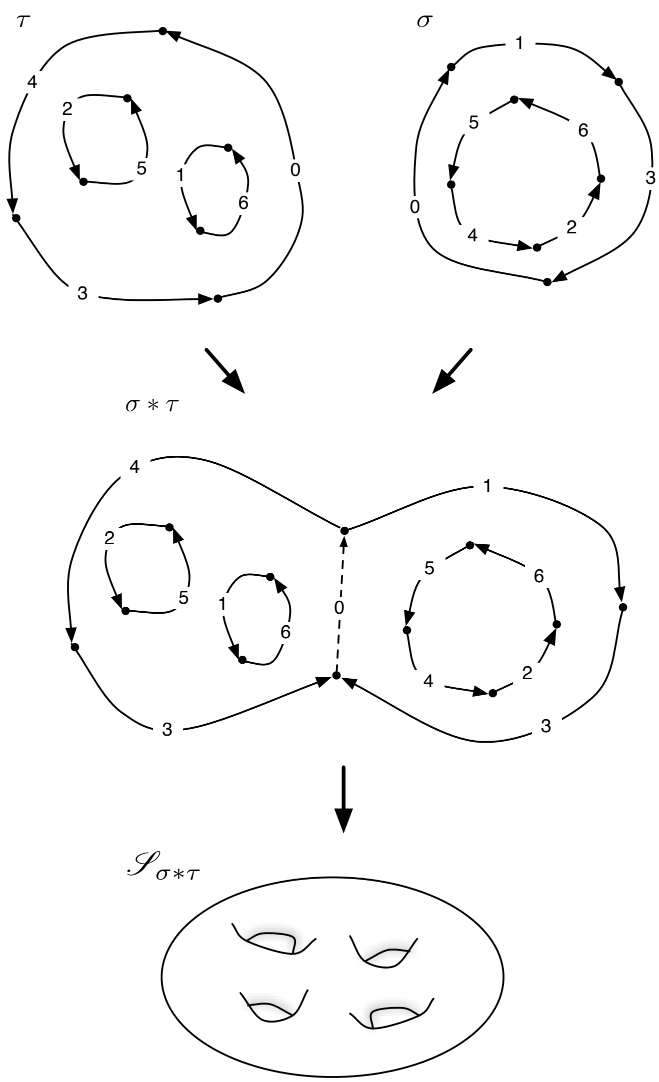


Fig. 5. Computing the distance through composition of surfaces.

We pose the problem of deciding relaxation when it is geometrically completed with "non-orientable" transformations. Because of the global features of Möbius and Klein rules, there is no hope to decide our extended notion of relaxation by stressing the theory of permutations. On the contrary, the geometrical procedure illustrated in these pages affords a global approach to relaxation which is guessed to be able to provide a neat solution for the problem at issue. This kind of result would be of interest in the specific field of surface morphing in which the notion of optimal strategy in topological evolution is absolutely central. Moreover, it could constitute a first step in the completely unexplored direction concerning the study of non-orientable features in non-commutative logical proofs.

Acknowledgments. I am grateful to Paul Ruet, who suggested me the idea of computing the distance by composing topological surfaces, and to the anonymous referee for his precious comments and suggestions.

References

1. V. M. Abrusci. Lambek calculus, cyclic linear logic, noncommutative linear logic: language and sequent calculus. In *Proofs and Linguistic Categories*. Cooperativa Libraria Universitaria Editrice Bologna: 21-48. 1997.
2. V. M. Abrusci and P. Ruet. Non-commutative logic I: the multiplicative fragment. *Annals of Pure and Applied Logic*, 101(1):29–64, 2000.
3. J.-M. Andreoli, G. Pulcini and P. Ruet. Permutative Logic. *Computer Science Logic*. Springer LNCS 3634: 184-199, 2005.
4. G. Bellin and A. Fleury. Planar and braided proof-nets for multiplicative linear logic with mix. *Archive for Mathematical Logic*, 37(5-6):309–325, 1998.
5. R. D. Carmichael. *Introduction to Theory of Groups of Finite Order.* Dover Publications Inc., 2000.
6. C. Gaubert. Two-dimensional proof-structures and the exchange rule. *Mathematical Structures in Computer Science*, 14(1):73–96, 2004.
7. J.-Y. Girard. Linear Logic: its syntax and semantics. *Advances in Linear Logic*, London Mathematical Society Lecture Note Series 222:1–42. Cambridge University Press, 1995.
8. W. S. Massey. *A basic course in algebraic topology.* Springer, 1991.
9. P.-A. Melliès. A topological correctness criterion for multiplicative non-commutative logic. In *Linear logic in computer science*, volume 316 of *London Mathematical Society Lecture Notes Series.* Cambridge University Press, 2004.
10. F. Métayer. Implicit exchange in multiplicative proofnets. *Mathematical Structures in Computer Science*, 11(2):261–272, 2001.
11. G. Pulcini. Permutative Additives and Exponentials. *Logic for Programming, Artificial Intelligence and Reasoning.* Springer LNAI 4790: 469-483, 2007.
12. G. Pulcini. Computing Surfaces via pq-Permutation. Chapter in the volume *Image Analysis - From Theory to Applications*, Research Publishing Services, 2008.
13. D.N. Yetter. Quantales and (non-commutative) linear logic. *Journal of Symbolic Logic*, 55(1), 1990.