Topology and its Applications 158 (2011) 748–760

Contents lists available at ScienceDirect

Topology and its Applications

www.elsevier.com/locate/topol

Classifying simple closed curve pairs in the 2-sphere and a generalization of the Schoenflies theorem ☆

Robert D. Franzosa [a,*], Ivan S. Gotchev [b], Daniel M. Look [c]

[a] Department of Mathematics and Statistics, University of Maine, 5752 Neville Hall, Orono, ME 04469-5752, United States
[b] Department of Mathematical Sciences, Central Connecticut State University, 1615 Stanley Street, New Britain, CT 06050, United States
[c] Department of Mathematics, Computer Science, and Statistics, St. Lawrence University, 23 Romoda Drive, Canton, NY 13617, United States

ARTICLE INFO

Article history:
Received 4 December 2008
Received in revised form 12 January 2011
Accepted 12 January 2011

MSC:
primary 05C10, 57N50
secondary 57M99, 57N35

Keywords:
Simple closed curve
Schoenflies theorem
Classification theory

ABSTRACT

A classification theory is developed for pairs of simple closed curves (A, B) in the sphere S^2, assuming that $A \cap B$ has finitely many components. Such a pair of simple closed curves is called an SCC-pair, and two SCC-pairs (A, B) and (A', B') are equivalent if there is a homeomorphism from S^2 to itself sending A to A' and B to B'. The simple cases where A and B coincide or A and B are disjoint are easily handled. The component code is defined to provide a classification of all of the other possibilities. The component code is not uniquely determined for a given SCC-pair, but it is straightforward that it is an invariant; i.e., that if (A, B) and (A', B') are equivalent and $\mathfrak{C}$ is a component code for (A, B), then $\mathfrak{C}$ is a component code for (A', B') as well. It is proved that the component code is a classifying invariant in the sense that if two SCC-pairs have a component code in common, then the SCC-pairs are equivalent. Furthermore code transformations on component codes are defined so that if one component code is known for a particular SCC-pair, then all other component codes for the SCC-pair can be determined via code transformations. This provides a notion of equivalence for component codes; specifically, two component codes are equivalent if there is a code transformation mapping one to the other. The main result of the paper asserts that if $\mathfrak{C}$ and $\mathfrak{C}'$ are component codes for SCC-pairs (A, B) and (A', B'), respectively, then (A, B) and (A', B') are equivalent if and only if $\mathfrak{C}$ and $\mathfrak{C}'$ are equivalent. Finally, a generalization of the Schoenflies theorem to SCC-pairs is presented.

1. Introduction

It follows by the Schoenflies theorem [4] that all simple closed curves in S^2 are equivalent in the sense that if A and A' are such simple closed curves, then there exists a homeomorphism $h : S^2 \to S^2$ mapping A to A'.

This result clearly does not extend to pairs (A, B) and (A', B') of simple closed curves in S^2.

The aim of this paper is to determine conditions under which one may infer whether or not a homeomorphism $h : S^2 \to S^2$ exists, mapping one pair of simple closed curves in S^2 to another.

We restrict ourselves to the case where the pairs of simple closed curves (A, B) under consideration are such that $A \cap B$ has finitely many components. We define an invariant that classifies all such pairs. The invariant is referred to as

☆ A preliminary version of much of the material in this paper was presented by the second-listed co-author at the Sixteenth Summer Conference on Topology and its Applications, City College of CUNY, New York, NY, USA, July 18–21, 2000.

* Corresponding author.
E-mail addresses: robert.franzosa@umit.maine.edu (R.D. Franzosa), gotchevi@ccsu.edu (I.S. Gotchev), dlook@stlawu.edu (D.M. Look).

0166-8641/$ – see front matter
doi:10.1016/j.topol.2011.01.016

a *component code*. This invariant is defined by combining local topological information at each intersection component of an SCC-pair with the relative ordering of these components around each simple closed curve.

Component codes are not uniquely determined for a pair of simple closed curves, but they are unique up to a component-code equivalence that we establish. Our main result, Theorem 6.3, states that if $\mathfrak{C}$ and $\mathfrak{C}'$ are component codes for simple closed curve pairs (A, B) and (A', B'), respectively, then there exists a homeomorphism $h : S^2 \to S^2$ mapping A to A' and B to B' if and only if $\mathfrak{C}$ is equivalent to $\mathfrak{C}'$. Also, in Theorem 7.2, we present a generalization of the Schoenflies theorem to a homeomorphism-extension result for pairs of simple closed curves that have a component code in common.

The problem of identifying the different ways that two (or more) sets can lie in relation to each other within an ambient space is central to the theory of spatial relations in the field of geographic information systems (GIS) and in part motivates the results presented herein. An invariant, similar to the component code, for the relation between pairs of disks in the plane is introduced in [1]. There is a large body of work on spatial relation theory within the GIS literature. The paper [2] is a seminal work on the topic.

Definition 1.1. An *SCC-pair* is a pair (A, B) of simple closed curves in S^2 that do not coincide, are not disjoint, and have finitely many components in their intersection. Two SCC-pairs (A, B) and (A', B') are said to be *equivalent* if there exists a homeomorphism $h : S^2 \to S^2$ mapping A to A' and B to B'.

Note that equivalence is straightforward for the two trivial types of pairs of simple closed curves excluded from the definition of SCC-pair. Specifically, if A and B are simple closed curves that coincide, and A' and B' are as well, then by the Schoenflies theorem there exists a homeomorphism $h : S^2 \to S^2$ mapping A to A' and B to B'. Also, if A and B are disjoint simple closed curves, and A' and B' are as well, then – by the annulus theorem [4] – in this case there also exists a homeomorphism $h : S^2 \to S^2$ mapping A to A' and B to B'.

2. Component codes

Let (A, B) be an SCC-pair. By component we refer to a component of the intersection of the simple closed curves A and B. In the situation that we are addressing, each component is either a point (called a *point component*) or an arc (called an *arc component*).

In a neighborhood of each component in S^2, the intersection between A and B topologically appears as one of the four possibilities shown in Fig. 1.

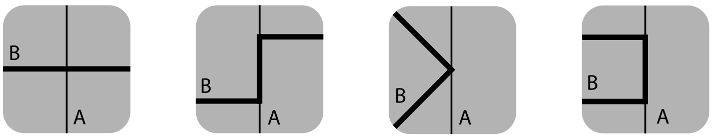


Fig. 1. Component types of an SCC-pair.

The first two of the possibilities shown in Fig. 1 we call *crossing components* and the latter two we call *touching components*. We refer to crossing or touching as the *type* of the component.

If two SCC-pairs are equivalent, then clearly there is a correspondence between the number of components and the type associated with each. On the other hand, as the example in Fig. 2 illustrates, knowing only the number of each type of component is not enough to classify the SCC-pairs. In the example, for each component type, the number of components of that type is the same in the two SCC-pairs (two crossing and two touching) but the SCC-pairs are clearly not equivalent.

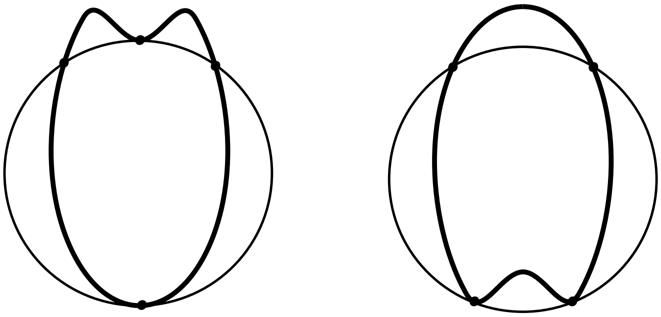


Fig. 2. Non-equivalent SCC-pairs with component types that coincide.

To define a classifying invariant, the information regarding the types of the components is "glued" together in a sequence describing the relative ordering of the components around each simple closed curve.

The point components and the endpoints of the arc components are called *contact points* for the SCC-pair. The contact points play an important role in constructing the classifying invariant that is the focus of our work.

For an SCC-pair (A, B), a *component code* is constructed as follows:

- Choose a direction around A and a contact point.
- Number the chosen contact point 0, and then number the remaining contact points in order around A in the chosen direction.
- Choose a direction around B.
- Begin at the component containing contact point 0, follow the chosen direction around B, and for each component:
  - if it is a point component, record the number and type of the component,
  - if it is an arc component, record the pair of numbers of the endpoints of the component, in order along B, as well as the type of the component.

The resulting sequence is called a *component code for the SCC-pair* (A, B). The contact point chosen in the first step is called the *initial contact point*, and the component containing it is called the *initial component*. The number assigned to each contact point is called the *label of the contact point*. Also, the number or pair of numbers recorded in the component code for each component is called the *label of the component*. Hence, each component code is in the form $(L_0T_0, \dots, L_{m-1}T_{m-1})$ where L_i is the label of the ith component (starting with the 0th) encountered around B, and T_i is the type of the component (C, for crossing, and T, for touching). We often blur the distinction between a contact point and its label, saying, for example, "contact point j", when what is meant is, "the contact point with label j".

Clearly, the component code is not uniquely determined; it depends on the chosen initial contact point and on the chosen directions on A and B. We call a particular choice of initial contact point and directions on A and B a *setup for* (A, B). Given a setup for (A, B), it uniquely determines a component code that we refer to as *the component code for the setup*.

We show in Section 6 that all possible component codes for an SCC-pair can be determined by one component code via mappings called *code transformations*. Thus for an SCC-pair one component code determines all component codes. Furthermore we show in Theorem 2.2 below that a component code determines (up to equivalence) the SCC-pair.

Before we proceed, we present in Fig. 3 two examples of component codes. On the left the SCC-pair has

$$\big(0C, 1C, [2,3]C, 8C, 7C, [4,5]T, 6C, 9T\big)$$

as a component code, and on the right the SCC-pair has

$$\big([12,0]T, 1C, [4,5]C, 3C, 2C, [7,6]C, 9C, 8T, 10C, 11C\big)$$

as a component code.

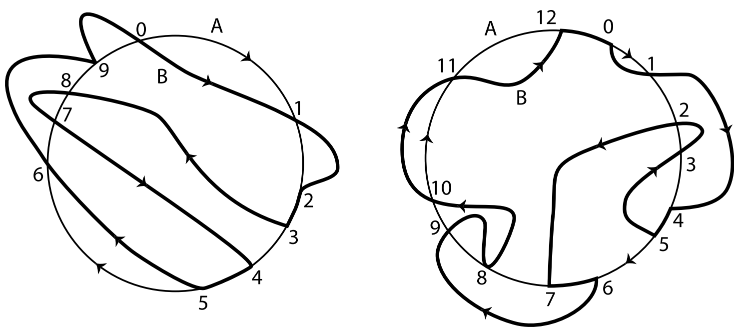


Fig. 3. Two examples of SCC-pairs.

It is clear that the component code is an invariant of SCC-pairs. We state that in the following:

Theorem 2.1. *If (A, B) and (A', B') are equivalent SCC-pairs and $\mathfrak{C}$ is a component code for (A, B), then $\mathfrak{C}$ is a component code for (A', B').*

One of our goals is to show that the component code is a classifying invariant. Specifically, we have:

Theorem 2.2 *(Preliminary classification theorem). If SCC-pairs (A, B) and (A', B') have a component code in common, then they are equivalent.*

We prove Theorem 2.2 in Sections 4 and 5.

In Section 6 we define transformations on the component codes that enable one to determine all possible component codes for a given SCC-pair from a particular component code for the pair. Via the transformations it is possible to determine whether or not two component codes represent equivalent SCC-pairs. Thus, we strengthen Theorems 2.1 and 2.2 with a result stating that two SCC-pairs are equivalent if and only if a given component code for one of the pairs is equivalent via code transformations to a given component code for the other pair.

In constructing the component code, one might question whether it would suffice to number just the components, rather than the contact points. The example in Fig. 4 shows that that is not the case. In Fig. 4 the illustrated SCC-pairs are not equivalent, but if we number just the components, and do so as illustrated, then we see that $(0C, 1C)$ is a "component code" for each of them. Thus, a sequence defined in this way is not going to distinguish between these two SCC-pairs and therefore does not classify SCC-pairs.

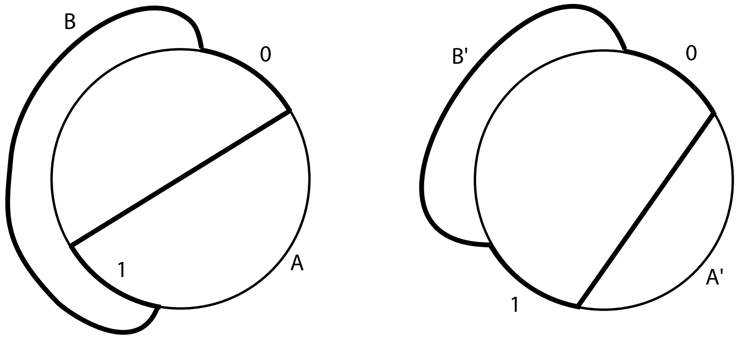


Fig. 4. Two SCC-pairs illustrating the need to number contact points and not just the components.

On the other hand, in the SCC-pair on the left in Fig. 4 if we number the contact points clockwise around A, starting at the top, then we see that $([0,1]C, [3,2]C)$ is a component code for (A, B). There is, however, no choice of setup such that $([0,1]C, [3,2]C)$ is a component code for (A', B').

Given this observation, and Theorem 2.2, we have found perhaps the simplest form for a classifying invariant for the SCC-pairs under consideration. It requires only the local topological type (*crossing* or *touching*) of each component and the relative orderings of the contact points around each simple closed curve.

3. Component direction codes

In this section we introduce the *component direction codes (CD codes)* associated to SCC-pairs. The CD codes include the information in the component codes and provide additional information about the chosen directions of travel along the simple closed curves through each component of their intersection. The information in the previously-introduced component code is sufficient for classifying SCC-pairs; the reason we introduce the CD codes is as an aid in establishing Theorem 2.2.

The main result in this section is Proposition 3.1 stating that if two SCC-pairs have a CD code in common then they are equivalent.

Proposition 3.1 is used in Section 4 in proving Theorem 2.2. There we show that if (A, B) and (A', B') are SCC-pairs with a component code in common, then either (A, B) and (A', B') have a CD code in common or there exists an SCC-pair (A'', B'') that has a CD code in common with (A, B) and that is equivalent to (A', B'). In the former case it follows from Proposition 3.1 that (A, B) and (A', B') are equivalent. In the latter case it follows from Proposition 3.1 that (A, B) and (A'', B'') are equivalent, thus implying that (A, B) and (A', B') are equivalent as well.

Let (A, B) be an SCC-pair. In a neighborhood of each component there is an arc in A entering the component (in the chosen direction on A) and an arc in A departing the component. As illustrated in two examples in Fig. 5, we denote these arcs by A_{in} and A_{out}, respectively. We similarly associate arcs B_{in} and B_{out} with each component.

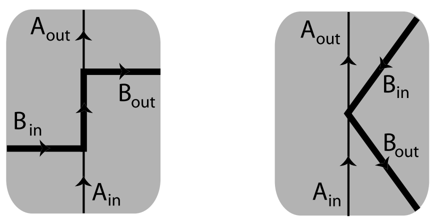
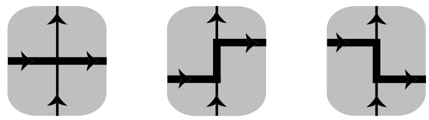
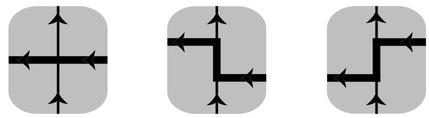
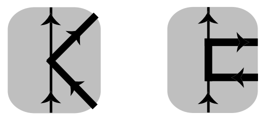
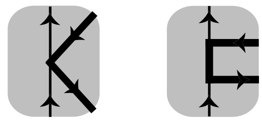
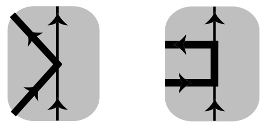
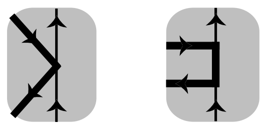


Fig. 5. Incoming and outgoing arcs at each component.

For the terms "left" and "right" to make sense in the definition of the CD code below, it is necessary to assume that we are viewing S^2 from a fixed "side" as the entries in the CD code are determined. Thus, in computing CD codes, we regard S^2 as a subspace of $\mathbb{R}^3$, and we assume we are viewing S^2 from the outside.

Note that if we include direction information along each simple closed curve, then in a neighborhood of each component in S^2 the intersection between A and B appears as one of the 14 different configurations shown in Table 1. In each case the thin line represents simple closed curve A and the thick line represents simple closed curve B. We refer to each of the six rows in the table as a *component direction type (CD type)* and each of the 14 different configurations as a *component direction configuration (CD configuration)*. In the first column, the expressions for the CD types describe how the path in the chosen direction along B passes through the corresponding component in relation to the path in the chosen direction along A. The CD types are discussed in more detail below.

Table 1
CD types and CD configurations.

CD Type	CD Configuration
CL **C**rossing from the **L**eft	
CR **C**rossing from the **R**ight	
TRS **T**ouching on the **R**ight, in the **S**ame direction	
TRO **T**ouching on the **R**ight, in the **O**pposite direction	
TLS **T**ouching on the **L**eft, in the **S**ame direction	
TLO **T**ouching on the **L**eft, in the **O**pposite direction	

The different CD types involve three different possible subtypes for each component:

- **Component subtype** – *crossing* or *touching* as in the component type defined previously.
- **Approach subtype** – *right* or *left*, determined by the side of A (relative to the chosen direction on A) from which the component is approached when traveling along B (in the chosen direction on B).
- **Touching direction subtype** – *same* or *opposite* indicating (roughly) whether travel along B through the component is in the same or opposite direction as travel along A (in the respective chosen directions). More precisely, for a touching component, consider the cyclic ordering of the arcs incident to the component going around it. If the arcs A_{in} and B_{in} are adjacent to each other in the ordering, then the component is of direction type *same*, otherwise it is of direction type *opposite*.

We construct the *component direction code (CD code)* in the same manner as the component code, only instead of recording the type (*crossing* or *touching*) of each component, we record the component direction type as described above. As with the component code, the CD code is not uniquely determined. As examples of CD codes, consider again the SCC-pairs in Fig. 3. The CD code for the pair on the left is

$$\big(0CL, 1CR, [2,3]CL, 8CR, 7CL, [4,5]TRS, 6CR, 9TLS\big)$$

and the CD code for the pair on the right is

$$\big([12,0]TRS, 1CR, [4,5]CL, 3CR, 2CL, [7,6]CR, 9CL, 8TRO, 10CR, 11CL\big).$$

Let an SCC-pair and CD code be given. Note that the label and CD type associated to a particular component determine which of the 14 CD configurations corresponds to the component. For example if the label and CD type is $[3,4]CR$ then the CD configuration appears as the second one in the second row of Table 1. Or if the label and CD type is $5TLO$ then the CD configuration appears as the first one in the last row.

In the component code the type of each component is independent of the chosen directions on each simple closed curve. In the CD code the CD type depends on the type of the component and on the chosen directions of travel.

Note that there are redundancies in the information in the CD code. For instance, if a touching component is labeled $[p, p+1]$, then its touching direction subtype must be "same", and if a touching component is labeled $[p, p-1]$, then its touching direction subtype must be "opposite". Furthermore, in Proposition 4.2 we show that in most instances the component code, along with the approach subtype of the initial component, determines the CD code. Keep in mind though, that the CD code is merely a support structure employed in proving Theorem 2.2. The motivation for the chosen organization of the CD code information is that it provides a simple means to determine the boundary of each component of the complement of an SCC-pair in S^2 (as demonstrated in Proposition 3.1, below).

Our goal now is to show that two SCC-pairs are equivalent if they have a CD code in common. We do that in Proposition 3.1 below, assuming that each SCC-pair does not have component code $(0T)$. The straightforward special case where the SCC-pairs have component code $(0T)$ is addressed in Section 5.

For an SCC-pair (A, B) observe that $A \cup B$ forms a graph $G_{A,B}$ in S^2. The contact points are the vertices, and the arcs that connect the contact points are the edges. We refer to the components of the complement of $G_{A,B}$ in S^2 as *regions*. Each component of $A \cap B$ lies in the boundary of a number of regions; we call these the *regions incident to the component*.

Consider a component of $A \cap B$. It is either a point component, labeled p, or an arc component, labeled $[p,q]$. In the first case we refer to the contact point p as both the *start contact point* and the *end contact point* for the component. In the latter case we refer to the contact point p as the *start contact point* and the contact point q as the *end contact point* for the component.

Proposition 3.1. *Let the SCC-pairs (A, B) and (A', B') have a CD code in common and be such that neither has component code $(0T)$. Then (A, B) and (A', B') are equivalent.*

Proof. Assume that the SCC-pairs (A, B) and (A', B') have the CD code $(L_0D_0, \dots, L_{m-1}D_{m-1})$ in common. Here L_i (as in the component code) represents the label of the component, and D_i represents the CD type of the component.

Furthermore assume that neither SCC-pair has component code $(0T)$. It can be easily shown that each intersection, $A \cap B$ and $A' \cap B'$, consists of more than one contact point.

We prove that there is a homeomorphism $h: S^2 \to S^2$ mapping A to A' and B to B'.

Note that there are m components associated to each of the SCC-pairs. Furthermore assume that there are n contact points (numbered $0, \dots, n-1$) associated to each of the SCC-pairs. In what follows, addition and subtraction on the labels of the contact points are calculated modulo n, but for notational convenience we do not indicate it so (i.e., we write $a+b$ rather than $(a+b) \bmod n$). Similarly addition and subtraction on the indices of the components are calculated modulo m but are not indicated so. The operation being performed should be clear from the context.

By an *oriented edge associated to* $G_{A,B}$ we mean an edge in the graph and an orientation that may or may not coincide with the chosen direction along either A or B in the setup for (A, B). For $i, j = 0, \dots, n-1$, we denote the oriented edge in A that begins at the contact point i and ends at the contact point j by ${}_iA_j$. Of course, here the only possibilities for j are $j = i+1$ or $j = i-1$. Similarly we let ${}_iB_j$ denote the oriented edge in B that begins at the contact point i and ends at the contact point j. The ordering of the contact points along B is recorded in the CD code, and therefore the oriented edges ${}_iB_j$ can be determined from the CD code. Note that each edge in $G_{A,B}$ has two oriented edges associated to it: ${}_iA_{i+1}$ and ${}_{i+1}A_i$ are different oriented edges associated to $G_{A,B}$, and if ${}_iB_j$ is an oriented edge associated to $G_{A,B}$, then ${}_jB_i$ is as well, but is different from ${}_iB_j$.

Since $A \cap B$ and $A' \cap B'$ consist of more than one point each, it follows that the graphs $G_{A,B}$ and $G_{A',B'}$ are 2-connected. Therefore each region is bounded by a simple closed curve that is a cycle of edges in the corresponding graph [3].

Let R be a region for $G_{A,B}$. By *the bounding cycle for* R we mean the cycle of oriented edges associated to $G_{A,B}$ that bounds R and is such that R lies to the right (relative to the orientation) of each edge in the cycle. We are assuming here that the cycle is independent of what might be listed as the starting edge; e.g. α, β, γ and β, γ, α would represent the same cycle.

We begin constructing the desired homeomorphism $h: S^2 \to S^2$ by defining it between the graphs $G_{A,B}$ and $G_{A',B'}$. The map on the vertices is defined by sending, for each $p = 1, \dots, n-1$, the contact point with label p in $G_{A,B}$ to the contact point with label p in $G_{A',B'}$. The map on the edges is defined by sending the oriented edges ${}_iA_{i+1}$ and ${}_iB_j$ homeomorphically to the oriented edges ${}_iA'_{i+1}$ and ${}_iB'_j$, respectively, preserving the orientation along the edges. Since SCC-pairs (A, B)

and (A', B') have the same CD code, it clearly follows that the result is a homeomorphism $h: G_{A,B} \to G_{A',B'}$ mapping A to A' and B to B'.

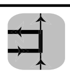
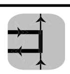
The task now is to extend h to a homeomorphism $h: S^2 \to S^2$. The Schoenflies theorem is used to do this. We show that the bounding cycle for each region is determined by the CD code. Thus, a natural correspondence is established between the regions for $G_{A,B}$ and those for $G_{A',B'}$. The Schoenflies theorem then allows us to extend $h: G_{A,B} \to G_{A',B'}$ to a homeomorphism between each pair of corresponding regions and therefore to a homeomorphism $h: S^2 \to S^2$.

Now, given a component C_k of $A \cap B$, consider the regions incident to it. The bounding cycle for each such region contains an oriented edge approaching C_k and an oriented edge departing C_k. Between these oriented edges there might also be an oriented edge running along C_k in the case that C_k is an arc component. We refer to this path of oriented edges approaching, possibly running along, and departing C_k as a *local boundary path* associated to C_k. The local boundary paths associated to C_k can be determined by the CD code as we now demonstrate.

From the CD code we can determine the end contact point in the component C_{k-1} and the start contact point in the component C_{k+1}; let the former be the contact point p and the latter be the contact point q.

Table 2 shows how the label on the component and its CD type determine the local boundary paths associated to the component. For each CD configuration, in the corresponding list of local boundary paths the first local boundary path corresponds to the region in the upper left or left half of the CD configuration diagram, and the rest of the local boundary paths correspond to the regions in cyclic order, going clockwise around the component.

Table 2
CD configuration diagrams and local boundary paths.

CD Type	Label	CD Configuration	Local Boundary Paths			
CL	j		${}_{j+1}A_jB_p$	${}_qB_jA_{j+1}$	${}_{j-1}A_jB_q$	${}_pB_jA_{j-1}$
	[j,j+1]		${}_{j+2}A_{j+1}A_jB_p$	${}_qB_{j+1}A_{i+2}$	${}_{j-1}A_jA_{j+1}B_q$	${}_pB_jA_{j-1}$
	[j+1,j]		${}_{j+2}A_{j+1}B_p$	${}_qB_jA_{j+1}A_{j+2}$	${}_{j-1}A_jB_q$	${}_pB_{j+1}A_jA_{j-1}$
CR	j		${}_{j+1}A_jB_q$	${}_pB_jA_{j+1}$	${}_{j-1}A_jB_p$	${}_qB_jA_{j-1}$
	[j,j+1]		${}_{j+2}A_{j+1}B_q$	${}_pB_jA_{j+1}A_{j+2}$	${}_{j-1}A_jB_p$	${}_qB_{j+1}A_jA_{j-1}$
	[j+1,j]		${}_{j+2}A_{j+1}A_jB_q$	${}_pB_{j+1}A_{j+2}$	${}_{j-1}A_jA_{j+1}B_p$	${}_qB_jA_{j-1}$
TRS	j		${}_{j+1}A_jA_{j-1}$	${}_qB_j\,A_{j+1}$	${}_pB_jB_q$	${}_{j-1}A_jB_p$
	[j,j+1]		${}_{j+2}A_{j+1}A_jA_{j-1}$	${}_qB_{j+1}A_{j+2}$	${}_pB_jA_{j+1}B_q$	${}_{j-1}A_jB_p$
TRO	j		${}_{j+1}A_jA_{j-1}$	${}_pB_jA_{j+1}$	${}_qB_jB_p$	${}_{j-1}A_jB_q$
	[j+1,j]		${}_{j+2}A_{j+1}A_jA_{j-1}$	${}_pB_{j+1}A_{j+2}$	${}_qB_jA_{j+1}B_p$	${}_{j-1}A_jB_q$
TLS	j		${}_{j+1}A_jB_q$	${}_{j-1}A_jA_{j+1}$	${}_pB_jA_{j-1}$	${}_qB_jB_p$
	[j,j+1]		${}_{j+2}A_{j+1}B_q$	${}_{j-1}A_jA_{j+1}A_{j+2}$	${}_pB_jA_{j-1}$	${}_qB_{j+1}A_jB_p$
TLO	j		${}_{j+1}A_jB_p$	${}_{j-1}A_jA_{j+1}$	${}_qB_jA_{j-1}$	${}_pB_jB_q$
	[j+1,j]		${}_{j+2}A_{j+1}B_p$	${}_{j-1}A_jA_{j+1}A_{j+2}$	${}_qB_jA_{j-1}$	${}_pB_{j+1}A_jB_q$

Thus, for example, the ${}_{j+1}A_jB_p$ entry in the first row of Table 2 indicates that the region (call it R) in the upper left corner in the associated CD configuration diagram has in its bounding cycle the oriented edges ${}_{j+1}A_j$ and ${}_jB_p$ in order.

Thus as we travel along A from contact point $j+1$ to contact point j and then along B to contact point p, the region R lies to the right of the path of travel.

Also the ${}_{j-1}A_jA_{j+1}B_q$ entry in the second row of Table 2 indicates that the region (call it R') in the lower right corner in the associated CD configuration diagram has in its bounding cycle the oriented edges ${}_{j-1}A_j$, ${}_jA_{j+1}$, and ${}_{j+1}B_q$ in order. Thus as we travel along A from contact point $j-1$ to contact point $j+1$, through contact point j, and then along B to contact point q, the region R' lies to the right of the path of travel.

Note that an oriented edge that corresponds to an arc component could be noted as both an oriented edge along A and as an oriented edge along B. In Table 2 we adopt the convention of always listing such an oriented edge as an oriented edge along A.

We claim that a CD code for (A, B) determines the bounding cycles for all of the regions for $G_{A,B}$.

To establish this claim we describe a procedure for determining the bounding cycles from a CD code.

To begin, note that a CD code determines all of the oriented edges associated to $G_{A,B}$. Each edge in $G_{A,B}$ has two oriented edges associated to it, and for each oriented edge there is exactly one region to its right.

Begin by choosing an oriented edge ${}_rA_s$ or ${}_rB_s$ for some $0 \leqslant r, s \leqslant n-1$. Denote the oriented edge by δ, and assume that R is the region to the right of δ. Since each region's bounding cycle must have an oriented edge that is not an arc component of $A \cap B$, we may assume that δ is not an arc component. We describe how to traverse the bounding cycle for R using information from the CD code. In this way we demonstrate how the bounding cycle can be determined from the CD code.

Begin by traversing δ from contact point r to contact point s. By examining the CD code, we can determine for which k component C_k coincides with or contains contact point s.

The label and CD type of component C_k can be read from the CD code. Furthermore, from the CD code we can determine the local boundary paths through C_k via Table 2. The oriented edge δ lies (first) in exactly one of these local boundary paths. It represents the approach into C_k, traversing the bounding cycle for R. The local boundary path tells us which oriented edges we must traverse to remain on the bounding cycle, approaching C_k along δ, passing through C_k, and departing C_k.

Let δ' be the oriented edge departing C_k. Assume that δ' runs from contact point r' to contact point s'. Repeat the above process to determine the oriented edges to traverse on the bounding cycle as it approaches, passes through, and departs the component coinciding with or containing contact point s'.

We continue the above process, traversing the bounding cycle, and we end when we encounter an oriented edge that we previously traversed. Since the bounding cycle is a simple closed curve, δ must be the first oriented edge that we encounter for a second time. Therefore the traverse ends when we return to δ. Clearly this traverse uniquely determines the bounding cycle for R.

To continue the procedure to determine all of the bounding cycles, pick an oriented edge in $G_{A,B}$ that is not an arc component and that is not in the bounding cycle determined above. Repeat the above process to determine the bounding cycle containing that oriented edge. Clearly, each oriented edge lies in a unique bounding cycle. In this way, we can continue determining bounding cycles until all of the oriented edges for $G_{A,B}$ have appeared in one. The result is the set of bounding cycles for all of the regions for $G_{A,B}$. This completes the proof of the claim.

Now it follows that each region for $G_{A,B}$ has a bounding simple closed curve of oriented edges uniquely determined by the CD code. Thus, there is a corresponding bounding simple closed curve and region in $G_{A',B'}$. The homeomorphism h (which so far is defined between $G_{A,B}$ and $G_{A',B'}$) maps each bounding simple closed curve of edges in $G_{A,B}$ to the corresponding one in $G_{A',B'}$. Thus, by the Schoenflies theorem, h can now be extended to map between corresponding regions and therefore to a homeomorphism $h : S^2 \to S^2$, mapping A to A' and B to B'. Thus, Proposition 3.1 has been established. □

The approach employed in the previous proof to extend the function h from a homeomorphism on $G_{A,B}$ to a homeomorphism on S^2 can be used to prove a generalization of the Schoenflies theorem to SCC-pairs. We discuss this further in Section 7.

4. Proof of the main case of the preliminary classification theorem

This section is devoted to proving Theorem 2.2, assuming that the SCC-pairs involved have neither $(0T)$ nor $(0T, 1T)$ as component codes. We address the special cases for SCC-pairs with component code $(0T)$ or $(0T, 1T)$ in Section 5.

Assume we have an SCC-pair (A, B) and a homeomorphism $f : S^2 \to S^2$. Then f naturally maps a setup for (A, B) to one for $(f(A), f(B))$; we call the latter *the induced setup on* $(f(A), f(B))$ resulting from f and the setup for (A, B).

Given a setup for (A, B), if the approach subtype for the initial component is left (right) then we say that the *setup has approach left* (*right*).

Let $f : S^2 \to S^2$ be a homeomorphism and (A, B) be an SCC-pair. Suppose we have a setup for (A, B). If the induced setup on $(f(A), f(B))$ has the opposite approach from the setup for (A, B), then we say that f *inverts* the setup for the SCC-pair (A, B). An example is illustrated in Fig. 6.

Proposition 4.1. *Given a setup for an SCC-pair (A, B), there exists a homeomorphism $f : S^2 \to S^2$ that inverts it.*

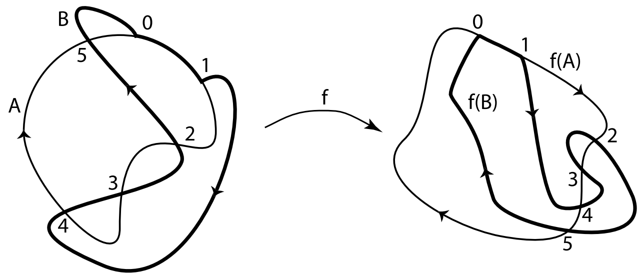


Fig. 6. The homeomorphism f inverts the setup for (A, B).

Proof. We compose three homeomorphisms to create the homeomorphism f.

- First, by the Schoenflies theorem there exists a homeomorphism g mapping S^2 onto itself in such a manner that A maps to the equator (i.e., considering S^2 as a subspace of $\mathbb{R}^3$, A maps to the set $\{(x, y, 0) \mid x^2 + y^2 = 1\}$).
- Next, follow g with the homeomorphism h on S^2 sending (x, y, z) to $(x, y, -z)$.
- Finally, map back by g^{-1}.

Note that the homeomorphism h fixes the equator (and therefore $g(A)$) and interchanges the northern and southern hemispheres. Clearly h inverts the setup for the SCC-pair $(g(A), g(B))$.

If the homeomorphism g inverts the setup for (A, B), then g^{-1} inverts the setup for $(h(g(A)), h(g(B)))$, and it then follows that $g^{-1} \circ h \circ g$ inverts the setup for (A, B). If g does not invert the setup for (A, B), then g^{-1} does not invert the setup for $(h(g(A)), h(g(B)))$, and again it follows that $g^{-1} \circ h \circ g$ inverts the setup for (A, B). In either case $f = g^{-1} \circ h \circ g$ inverts the setup for (A, B). □

The following proposition is the main step in establishing the desired result in this section. It indicates that if a component code is known, along with the approach of the associated setup, then the CD code for the setup is uniquely determined.

Proposition 4.2. *Assume that (A, B) is an SCC-pair and that a setup is given. Let $\mathfrak{C}$ be the associated component code, and assume that $\mathfrak{C}$ equals neither $(0T)$ nor $(0T, 1T)$. Then the approach of the setup and component code $\mathfrak{C}$ uniquely determine the CD code for the setup.*

Proof. Let (A, B) and $\mathfrak{C}$ be as in the statement of the theorem. If $\mathfrak{C} = (L_0T_0, \ldots, L_{m-1}T_{m-1})$, then the corresponding CD code must be in the form $(L_0D_0, \ldots, L_{m-1}D_{m-1})$. We show, for $k = 0, \ldots, m-1$, how we can determine the CD type, D_k, of the kth component from the approach of the setup and the component code $\mathfrak{C}$.

For each component, in order to determine its CD type we need to determine its component subtype (*crossing* or *touching*), approach subtype (*right* or *left*), and – where appropriate – touching direction subtype (*same* or *opposite*).

The component subtype can be directly determined from the component code. Specifically, for the kth component the component subtype in D_k is given by T_k.

Next, with the approach of the setup given, the approach subtype in D_0 is known because the approach of the setup is defined to be the approach subtype of the initial component. Then the approach subtypes of the remaining components are easily determined from the approach subtype in D_0 and the component code $(L_0T_0, \ldots, L_{m-1}T_{m-1})$. Specifically, given the approach subtype of the kth component, if $T_k = \textit{touching}$, then the approach subtype of the $(k+1)$st component is the same as that of the kth component, and if $T_k = \textit{crossing}$, then the approach subtype of the $(k+1)$st component is the opposite of that of the kth component. Basically the idea here is that as we traverse B, the approach subtypes of the components do not change as long as we remain on the same side of A going through touching components, but after we cross A with a crossing component, the approach subtype of the next component takes on the opposite value.

Finally, we show how the touching direction subtype for each touching component can be determined. First, note that if the component is an arc component, then the touching direction subtype can be determined from the label of the component in $\mathfrak{C}$. Specifically, if the label is $[p, p+1]$ then the touching direction subtype is *same*, and if the label is $[p, p-1]$ then the touching direction subtype is *opposite*.

Now let us consider the situation where the component has subtype *touching* and is a point component with label p. From $\mathfrak{C}$ we can determine the end contact point in the component that lies prior to contact point p as B is traversed in its chosen direction. Assume that that contact point is p^-. Furthermore we can determine the start contact point in the component that lies just after contact point p; assume that it is labeled p^+. Since the component code $\mathfrak{C}$ equals neither $(0T)$ nor $(0T, 1T)$, it follows that p^-, p, and p^+ are distinct. The situation appears as in Fig. 7.

Note that since A is a simple closed curve in S^2, the two components of the complement of A are topologically the same, and therefore in Fig. 7 whether we depict B_{in} and B_{out} as “inside” A or “outside” A does not matter. What does matter, though, is that since p is a touching component, it follows that B_{in} and B_{out} are on the same side of A.

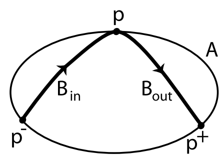


Fig. 7. Configuration for a point component p of type touching.

Finally, we can determine the touching direction subtype for the component with label p from the values of p^-, p, and p^+ (and therefore from their relative position on A). In particular, if $(p^+ - p) \bmod n < (p^- - p) \bmod n$, then in traversing A in the chosen direction, starting at p, we encounter p^+ prior to p^-. It follows that in Fig. 7 the chosen direction on A is clockwise, and therefore the CD configuration at p is as in Fig. 8(a). Thus the component with label p has touching direction subtype *same*.

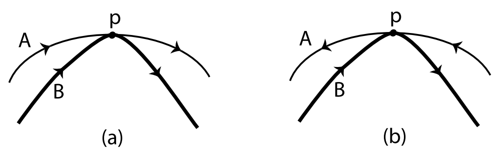


Fig. 8. On the left p has subtype *same*, and on the right p has subtype *opposite*.

Similarly it follows that if $(p^- - p) \bmod n < (p^+ - p) \bmod n$, then the component with label p has touching direction subtype *opposite* as in Fig. 8(b).

Thus we have shown that the approach of a setup and the associated component code uniquely determine the CD code for the setup, completing the proof of Proposition 4.2. □

Propositions 3.1, 4.1, and 4.2 combined yield the desired result in this section.

Theorem 4.3 *(Main case of the preliminary classification theorem). Assume that $\mathfrak{C}$ is a component code for the SCC-pairs (A, B) and (A', B'). Further assume that $\mathfrak{C}$ equals neither $(0T)$ nor $(0T, 1T)$. Then (A, B) and (A', B') are equivalent.*

Proof. Assume that setups for (A, B) and (A', B') are given and that $\mathfrak{C}$ is the associated component code in each case.

If the setups have the same approach, then by Proposition 4.2 the CD code associated to the setup on (A, B) equals the CD code associated to the setup on (A', B'). Therefore by Proposition 3.1 it follows that (A, B) and (A', B') are equivalent.

If the setups have the opposite approach, then by Proposition 4.1 there exists a homeomorphism $f : S^2 \to S^2$ that inverts the setup for (A', B'). Let $(A'', B'') = (f(A'), f(B'))$. The induced setup for (A'', B'') has the same approach and same associated component code $\mathfrak{C}$ as for the given setup for (A, B). As above, it follows that (A, B) and (A'', B'') are equivalent. Therefore, since (A', B') and (A'', B'') are equivalent, it follows that (A, B) and (A', B') are equivalent in this case as well. □

5. Proof of the special cases of the preliminary classification theorem

In this section we address Theorem 2.2 for the special cases where the SCC-pairs (A, B) and (A', B') both have $(0T)$ or both have $(0T, 1T)$ as component codes. As in the proof of Proposition 3.1, we derive a homeomorphism $h : S^2 \to S^2$, mapping A to A' and B to B', by first defining a homeomorphism from a graph containing $A \cup B$ to a graph containing $A' \cup B'$ and then extending to a homeomorphism on S^2. The results here are not difficult; they just need to be handled in a manner different from the general approach presented above.

To begin, assume that (A, B) and (A', B') both have component code $(0T, 1T)$. In this case the graph $G_{A,B}$ consists of two vertices (labeled 0 and 1) and four edges between them, with two of the edges making up A and two of the edges making up B, as shown in Fig. 9. The graph $G_{A',B'}$ is structured similarly. Note that these graphs are 2-connected, and therefore each associated region in S^2 is bounded by a simple closed curve that is a cycle of edges in the corresponding graph.

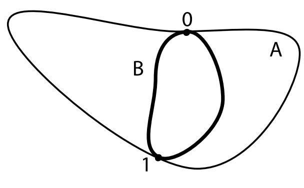


Fig. 9. SCC-pair (A, B) with component code $(0T, 1T)$.

To define the desired homeomorphism between $G_{A,B}$ and $G_{A',B'}$, begin by labeling the edges in $G_{A,B}$ as A_1, A_2, B_1, B_2 in such a way that edges A_1 and A_2 make up A, edges B_1 and B_2 make up B, and the four edges appear in consecutive order when cycling around the vertex labeled 0 (as illustrated in Fig. 10). Similarly label the edges in $G_{A',B'}$ as A'_1, A'_2, B'_1, B'_2.

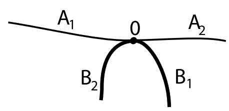


Fig. 10. Labeling the edges in $G_{A,B}$.

Now define a homeomorphism $g : G_{A,B} \to G_{A',B'}$ by sending the vertices labeled 0 and 1 in $G_{A,B}$ to the respective vertices in $G_{A',B'}$ and by sending the edges A_1, A_2, B_1, and B_2 to A'_1, A'_2, B'_1, and B'_2, respectively. With the homeomorphism as defined, it is easy to see that each simple closed curve in $G_{A,B}$ that makes up the boundary of a region associated to $G_{A,B}$ is sent to a corresponding simple closed curve in $G_{A',B'}$ making up the boundary of a region associated to $G_{A',B'}$. Thus by the Schoenflies theorem, we can extend g over each of the regions associated to $G_{A,B}$, thereby obtaining a homeomorphism $h : S^2 \to S^2$, mapping A to A' and B to B'.

Therefore if SCC-pairs (A, B) and (A', B') both have $(0T, 1T)$ as a component code, then they are equivalent.

Now assume that (A, B) and (A', B') both have component code $(0T)$. In this case the graphs $G_{A,B}$ and $G_{A',B'}$ are topologically figure eights. So a figure-eight version of the Schoenflies theorem is all that is needed to assert the existence of an equivalence between (A, B) and (A', B'). Such an extension of the Schoenflies theorem is straightforward to accomplish; we outline how here.

There are three regions associated to the graph $G_{A,B}$: one bounded by A, one bounded by B, and one bounded by both A and B. There exist arcs in S^2 that begin at a point in $A - B$, end at a point in $B - A$, and otherwise lie in the region that is bounded by both A and B (Fig. 11). Pick such an arc, α, and let G be the graph formed by $A \cup B \cup \alpha$.

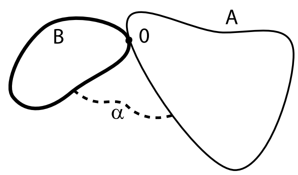


Fig. 11. Arc α runs from simple closed curve A to simple closed curve B.

We similarly construct a graph G' containing $A' \cup B'$ and an arc in S^2 beginning at a point in $A' - B'$ and ending at a point in $B' - A'$. The graphs G and G' are 2-connected, and therefore the regions in S^2 associated to each graph are bounded by simple closed curves in the graphs. We can easily define a homeomorphism $g : G \to G'$, sending A to A', B to B', and each simple closed curve in G that bounds a region associated to G to a simple closed curve in G' that bounds a region associated to G'. Such a homeomorphism then, by the Schoenflies theorem, can be extended to a homeomorphism $h : S^2 \to S^2$ providing the desired equivalence between the SCC-pairs (A, B) and (A', B').

Therefore if the SCC-pairs (A, B) and (A', B') both have $(0T)$ as a component code, then they are equivalent.

6. Code transformations

In this section we define transformations on the component codes enabling one to determine all possible component codes for a given SCC-pair from a particular component code for the SCC-pair. The significance of the results that we present here is that via these transformations it is possible to determine whether or not two component codes represent equivalent SCC-pairs. Thus, we strengthen Theorem 2.2 with a result stating that two SCC-pairs are equivalent if and only if a given component code for one of the pairs is “equivalent” via code transformations to a given component code for the other.

Let $\mathfrak{C} = (L_0T_0, \dots, L_{m-1}T_{m-1})$ be a component code for (A, B) for a particular setup. Furthermore, assume that there are n contact points associated to (A, B).

As above, addition and subtraction on the labels of the contact points are calculated modulo n but are not indicated so, and addition and subtraction on the indices of the components are calculated modulo m but are not indicated so.

We begin with some notation. Each label L_k is either a single value, p, or a pair of values, $[p, q]$. For each j such that $0 \leqslant j \leqslant n-1$, define $L_k - j$ to be $p - j$ in the first case and $[p - j, q - j]$ in the second case. Similarly define $j - L_k$. Further, define $S(L_k)$ to be p in the first case and $[q, p]$ in the second. S switches the order of the values on a label associated to an arc component.

Now we wish to examine the effect on the component code $\mathfrak{C}$ resulting from changing the initial contact point, the direction around A, or the direction around B. First note that such changes do not change the type of any component; it is only the labels and the ordering of the terms L_kT_k in the component code that are affected.

Consider the situation where we change the direction around A, maintaining the same initial contact point and direction around B. In this case the labels change as a result of the new numbering around A, but the location and type of each component in the component-code m-tuple is unchanged. Clearly a contact point previously labeled j is now labeled $n-j$.

Thus define the code transformation

$$D_1\big((L_0T_0,\dots,L_{m-1}T_{m-1})\big)=\big((n-L_0)T_0,(n-L_1)T_1,\dots,(n-L_{m-1})T_{m-1}\big).$$

If $\mathfrak{C}$ is a component code for a setup for (A,B), and $\mathfrak{C}'$ is the component code for the setup with the same choice of initial contact point and direction around B, and the opposite choice of direction around A, then clearly $\mathfrak{C}'=D_1(\mathfrak{C})$.

Next consider the case where we change the direction around B, maintaining the same initial contact point and direction around A. Clearly the resulting component code is obtained from $\mathfrak{C}$ by beginning with the same initial component, listing the terms L_kT_k in the opposite order, and switching the order of the values associated with each arc component.

Thus, define the code transformation

$$D_2\big((L_0T_0,\dots,L_{m-1}T_{m-1})\big)=\big(S(L_0)T_0,S(L_{m-1})T_{m-1},S(L_{m-2})T_{m-2},\dots,S(L_1)T_1\big).$$

If $\mathfrak{C}$ is a component code for a setup for (A,B), and $\mathfrak{C}'$ is the component code for the setup with the same choice of initial contact point and direction around A, and the opposite choice of direction around B, then clearly $\mathfrak{C}'=D_2(\mathfrak{C})$.

Finally, we consider the situation where we change the initial contact point while maintaining the same directions around A and B. Assume that the new initial contact point was the contact point originally labeled j, $0\leqslant j\leqslant n-1$. A contact point originally labeled p is now labeled $p-j$. So the labels of each component change from L_k to L_k-j as a result of the new numbering around A.

Furthermore, the location of each term in the component-code m-tuple changes since the new initial contact point results in a new initial component. Note that there exists a unique $k\in\{0,\dots,m-1\}$ such that B_k equals either j, $[j\pm1,j]$, or $[j,j\pm1]$. Let $c(j)=k$. What was originally the kth component is now the initial component in the new component code, what was originally $(k+1)$st is now the second, etc.

The effect of "rotating" the initial contact point to the jth contact point is captured in the code transformation R_j defined as follows:

$$R_j\big((L_0T_0,\dots,L_{m-1}T_{m-1})\big)=\big((L_{c(j)}-j)T_{c(j)},(L_{c(j)+1}-j)T_{c(j)+1},\dots,(L_{c(j)+m-1}-j)T_{c(j)+m-1}\big)$$

where $c(j)$ is the unique value such that $L_{c(j)}=j$, $[j\pm1,j]$, or $[j,j\pm1]$.

If $\mathfrak{C}$ is a component code for a setup for (A,B), and $\mathfrak{C}'$ is the component code for the setup with the same choice of directions around A and B, and initial point equal to contact point j in the given setup, then clearly $\mathfrak{C}'=R_j(\mathfrak{C})$.

Now, let $\mathfrak{C}$ be a component code for a particular setup for (A,B). All other component codes for the SCC-pair can be obtained by changing the choice of initial contact point, direction around A, or direction around B. Thus all other component codes can be obtained by compositions of the transformations D_1, D_2, and R_j acting on $\mathfrak{C}$. Conversely if $\mathfrak{C}'$ is obtained from $\mathfrak{C}$ by a composition of these transformations, then $\mathfrak{C}'$ is clearly a component code for (A,B).

Definition 6.1. Two component codes are *equivalent* if one can be obtained from the other via a composition of the transformations D_1, D_2, R_j, where $0\leqslant j\leqslant n-1$.

It can be shown that any finite composition of the transformations D_1, D_2, R_j reduces to one of the following types: the identity, D_1, D_2, R_j, $D_1\circ D_2$, $D_1\circ R_j$, $D_2\circ R_j$, or $D_1\circ D_2\circ R_j$. We do not pursue that result here as it is not needed.

It is not difficult to show that the equivalence defined above is an equivalence relation.

Note that this notion of equivalence is defined via the transformations on component codes, independent of any underlying SCC-pair. We have, however, the following theorem whose proof is straightforward.

Theorem 6.2. *Two component codes are component codes for the same SCC-pair if and only if the component codes are equivalent.*

This now brings us to the main result of this paper:

Theorem 6.3 *(Main classification theorem). Let $\mathfrak{C}$ and $\mathfrak{C}'$ be component codes for SCC-pairs (A,B) and (A',B'), respectively. Then (A,B) and (A',B') are equivalent if and only if $\mathfrak{C}$ and $\mathfrak{C}'$ are equivalent.*

Proof. If (A,B) and (A',B') are equivalent SCC-pairs, then by Theorem 2.1, $\mathfrak{C}$ and $\mathfrak{C}'$ are both component codes for both SCC-pairs (A,B) and (A',B'). Consequently, by Theorem 6.2, $\mathfrak{C}$ and $\mathfrak{C}'$ are equivalent.

On the other hand, if $\mathfrak{C}$ and $\mathfrak{C}'$ are equivalent component codes, then by Theorem 6.2, they are both component codes for both SCC-pairs (A,B) and (A',B'). Theorem 2.2 then implies that the SCC-pairs (A,B) and (A',B') are equivalent. □

7. A generalization of the Schoenflies theorem

In this section we present a generalization of the Schoenflies theorem to SCC-pairs. The Schoenflies theorem indicates that if $h : A \to A'$ is a homeomorphism between simple closed curves in S^2 then h extends to a homeomorphism from S^2 to itself. In our case, we provide conditions under which a homeomorphism between SCC-pairs can be extended to S^2.

To begin we have the following:

Definition 7.1.

- A *homeomorphism between SCC-pairs* (A, B) and (A', B') is a homeomorphism $h : A \cup B \to A' \cup B'$ mapping A to A' and B to B'.
- A homeomorphism h between SCC-pairs (A, B) and (A', B') is *component-code preserving* if the SCC-pairs have a component code $\mathfrak{C}$ in common and if the setups for $\mathfrak{C}$ in (A, B) and (A', B') are such that h maps contact point j in A to contact point j in A' for each label j assigned to the contact points.

Except for two special cases, component-code preserving homeomorphisms extend to S^2, as indicated by the following generalization of the Schoenflies theorem:

Theorem 7.2. *If h is a component-code preserving homeomorphism between SCC-pairs (A, B) and (A', B'), and neither $(0T)$ nor $(0T, 1T)$ is a component code for the SCC-pairs, then h extends to a homeomorphism from S^2 to itself.*

We discuss how the proof of Theorem 7.2 proceeds similar to previous results in this paper. Let the SCC-pairs (A, B) and (A', B') have a component code $\mathfrak{C}$ in common (and equal to neither $(0T)$ nor $(0T, 1T)$). Further, assume that h is a component-code preserving homeomorphism between (A, B) and (A', B'). Finally, assume that for each j, the homeomorphism h maps contact point j in A to contact point j in A' in the setups for $\mathfrak{C}$.

If the setups for $\mathfrak{C}$ in (A, B) and (A', B') have the same approach, then by Propositions 4.2 and 3.1 the SCC-pairs are equivalent. As is done in the proof of Proposition 3.1, the equivalence can be established by using the Schoenflies theorem to extend h to each component of the complement of $A \cup B$, thereby yielding the desired homeomorphism from S^2 to itself.

If the setups for $\mathfrak{C}$ in (A, B) and (A', B') have the opposite approach, then by Proposition 4.1 there exists a homeomorphism $f : S^2 \to S^2$ that inverts the setup for (A', B'). If we let $(A'', B'') = (f(A'), f(B'))$, then the induced setup for (A'', B'') has the same approach and same associated component code $\mathfrak{C}$ as for the given setup for (A, B). The function $f|_{A' \cup B'} \circ h$ is a component-code preserving homeomorphism between (A, B) and (A'', B''), and, as above, can be extended to a homeomorphism H from S^2 to itself. The homeomorphism $f^{-1} \circ H$ is the desired extension of h.

If we relax any one of the assumptions in Theorem 7.2 that h is component-code preserving, that $\mathfrak{C} \neq (0T)$, or that $\mathfrak{C} \neq (0T, 1T)$, then it is straightforward to find examples of homeomorphisms between $A \cup B$ and $A' \cup B'$ that do not extend to S^2. For example, in Fig. 12, if $h : A \cup B \to A' \cup B'$ is a homeomorphism mapping A clockwise to A' and mapping B counterclockwise to B', then h is a component-code preserving homeomorphism that does not extend to a homeomorphism on S^2.

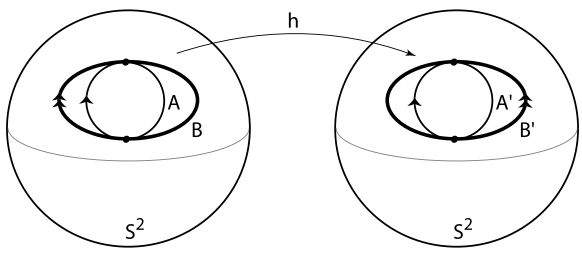


Fig. 12. A component-code preserving homeomorphism that does not extend to S^2.

Acknowledgements

The results in this paper arose out of conversations with Max Egenhofer of the Department of Spatial Information Science and Engineering at the University of Maine.

References

[1] M. Egenhofer, R. Franzosa, Point-set topological spatial relations, International Journal of Geographical Information Systems 8 (1991) 161–174.
[2] M. Egenhofer, R. Franzosa, On the equivalence of topological relations, International Journal of Geographical Information Systems 9 (1995) 133–152.
[3] B. Mohar, C. Thomassen, Graphs on Surfaces, The Johns Hopkins University Press, 2001.
[4] E. Moise, Geometric Topology in Dimensions 2 and 3, Springer-Verlag, 1977.