THE LABELLED PEER CODE
FOR KNOT AND LINK DIAGRAMS

26th February, 2015

A **labelled peer code** is a descriptive syntax for a diagram of a knot or link on a two dimensional sphere. The syntax is able to describe classical, virtual, welded or flat knot and link diagrams.

1 Peer Codes

Let D be an oriented diagram of a knot or link with k components and let $\mathcal{I}(D)$ be the immersion of D in the plane. Thus $\mathcal{I}(D)$ is a 4-regular plane graph with n vertices and $2n$ edges, sometimes referred to as a **shadow**, and it inherits an orientation from D. Each vertex in $\mathcal{I}(D)$ has two incoming edges and two outgoing edges, with respect to the orientation. The two incoming edges are called *peers*, or *peer edges*. We shall refer to the image of each component of D in $\mathcal{I}(D)$ as a component of $\mathcal{I}(D)$.

Lemma The image of each component of D in $\mathcal{I}(D)$ contains an even number of edges.

Proof Suppose C is a component $\mathcal{I}(D)$. Seifert smooth any self intersections of C to form C', which is a union of disjoint, oriented simple closed curves. There must be an even number of intersections of each component of C' with $\mathcal{I}(D) - C$ since, by the Jordan Curve Theorem, as we trace every other component of $\mathcal{I}(D)$ following its orientation, for each time that we enter the interior of a component of C' we must also leave that interior. Thus C' contains an even number of edges, and since re-instating the self instersections adds a pair of edges, so does C.

Lemma The regions of $\mathcal{I}(D)$'s complement in the plane may be two-coloured in a chessboard fashion.

Proof This is a special case of a more general result concerning planar graphs G whose vertices all have even valency. If G is such a graph, regarded as lying in S^2, and G' is its dual graph, then the regions of G' all have an even number of edges. By choosing a colour for the vertex of one region arbitrarily and extending the colouring of the vertices of G' by alternating the colouring around the regions of G' in turn it is possible to two-colour the vertices of G' in a consistent manner. This two-colouring of the vertices of G' corresponds to a two-colouring of the regions of G.

Choose an initial component of D and a basepoint b for that component that does not map to a vertex in $\mathcal{I}(D)$. Number the edges in $\mathcal{I}(D)$ consecutively from zero starting at the edge containing image of b and following the path determined by the orientation of D. This results in crossings of $\mathcal{I}(D)$ whose incoming edges have both been numbered having one peer assigned an even number and the other peer an odd number.

Now proceed successively through the remaining components of D in a similar manner, each time choosing the next component so that its image in $\mathcal{I}(D)$ involves a crossing that has already had a number assigned to one of its incoming edges. Extend the numbering to the corresponding edges in $\mathcal{I}(D)$ choosing a basepoint on the next component so that crossings whose incoming edges have both been numbered have one peer assigned an even number and the other peer an odd number. It is always possible to choose such a basepoint: the following argument is due to Roger Fenn.

Lemma Let S be a shadow in general position. Then the edges of S can be coloured by two colours, orange and emerald, so that at each crossing the incoming edges have different colours, as do the outgoing edges. Moreover the colours change as the edges cross the crossing.

Proof Colour the regions black and white chessboard fashion. Now orient the crossings so that the incoming edges are on the left and the outgoing edges are on the right. If the region above is black colour the top edges orange and the bottom edges emerald. If the region above is white, do the opposite. It is easy to check that this defines a coherent colour for each edge which satisfies the above.

Corollary The edges of a shadow in general position with n crossings can be labelled consecutively respecting the orientation with the integers from 1 to 2n such that at any crossing the incoming edges are odd and even; likewise the outgoing edges.

Proof Colour the edges orange and emerald as in the above lemma. Pick an orange edge and label it 1. Continue past the next crossing and label the next edge, coloured with emerald, 2. Continue in this fashion until all the integers from 1 to 2k say are used for this component. For the second component chose an orange edge and label it 2k+1, etc

The numbering of edges in $\mathcal{I}(D)$ induces a unique numbering of the vertices $0, \ldots, n-1$ by assigning the vertex at which edge $2i$ terminates the number i.

Definition 1.1 We shall refer to the edge numbered $2i$ terminating at vertex i as the **naming edge** for that vertex.

Note also that the numbering of the edges of $\mathcal{I}(D)$ determines a permutation ρ on n elements as follows. At each vertex i the incoming edges are numbered $2i$ and $2j-1$ for some $j \in \{0, \ldots, n-1\}$ where we count edges modulo $2n$. Define $\rho(i) = j$. This permutation allows us to enumerate peer codes, as described below.

From the numbering of the edges of $\mathcal{I}(D)$ we may write a list of the odd numbered peers in the order determined by the vertex numbering. We separate this list into the peers of those naming edges that are associated with the same component of D

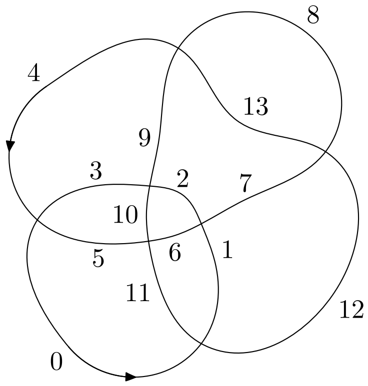

Figure 1.

For example, for the immersion and edge numbering shown in Figure 1, the list of odd peers is

$$11\ 9,\ \ 3\ 1\ 13\ 5\ 7$$

1.1 Type I and Type II crossings

There are two possibilites for the relative numbering of incoming edges at a vertex of $\mathcal{I}(D)$, as shown in Figure 2.

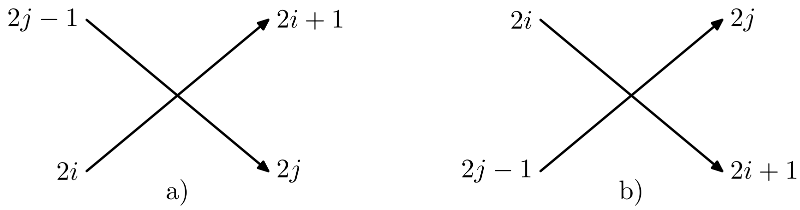

Figure 2.

Definition 1.2 A crossing of the type shown in Figure 2 a) is called a **Type I** crossing and a crossing of the type shown in Figure 2 b) is called a **Type II** crossing.

The list of odd peers may be supplemented to record the type of each crossing by writing each odd peer associated with a Type I crossing as a negative number.

For the immersion in Figure 1 we get

$$-11\ 9,\ -3\ 1\ \ -13\ 5\ \ -7$$

we refer to this code as a *peer code*.

2 Labelled Peer Codes

We may describe a knot or link diagram D fully by giving its peer code together with a set of labels that describe each crossing. For classical crossings we assign the label $+$ if the naming edge in $\mathcal{I}(D)$ forms part of the over-arc of the crossing and the label $-$ if it forms part of the under-arc. For virtual or welded crossings we assign the label $*$ and for flat crossings we assign the label $\#$.

Definition 2.1 A **labelled peer code** for a diagram D is a peer code for D together with a set of labels, one for each crossing. It is written as the peer code followed by a '/' character, followed in turn by the labels. The labels appear in the order induced on the vertices of $\mathcal{I}(D)$ by the numbering of its edges.

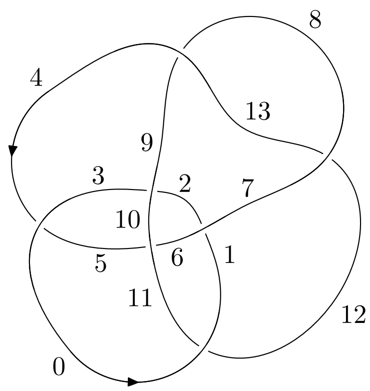

Figure 3.

Thus, the labelled peer code for the link and numbering shown in Figure 3 is

$$-11\ 9,\ -3\ 1\ \ -13\ 5\ \ -7\ /+\ -\ -\ +\ -\ +\ -$$

For the purposes of distinguishing labelled peer codes from other codes when using a computer the peer code will be enclosed in square brackets, as follows:

$$[-11\ 9,\ -3\ 1\ \ -13\ 5\ \ -7]\ /+\ -\ -\ +\ -\ +\ -$$

2.1 Realizable Peer Codes

Although patently the peer code for a link diagram is not unique, an interesting question is to ask when a given peer code corresponds to a realizable diagram. This may be answered by noticing that a connected immersion $\mathcal{I}(D)$ determines a cellular decomposition of S^2 and so by Euler's theorem the number of components of $S^2 - \mathcal{I}(D)$ is $n+2$.

Definition 2.2 For each even numbered edge e in a connected immersion $\mathcal{I}(D)$ there is a sequence of edges $e_0, \ldots, e_k$ with $e = e_0 = e_k$ called the **left turning cycle** obtained by turning left at each crossing we encounter as we trace around $\mathcal{I}(D)$ starting by moving along e following the orientation of $\mathcal{I}(D)$. Similarly we define the **right turning cycle** for e as the corresponding sequence obtained by always turning right. We define left and right turning cycles for odd numbered edges in the same way but require that we start by moving along the edge against the orientation of $\mathcal{I}(D)$.

Clearly every edge in a left (right) turning cycle will determine the same left (right) turning cycle.

Given a peer code, we are able to determine unambiguously the edge we encounter when turning left or right at a crossing, whether we have arrived following the orientation or not (see Figure 2). We are therefore able to determine whether the peer code is connected or not and, if so, may determine $\mathcal{L}$ the set of distinct left turning cycles, $\mathcal{R}$ the set of distinct right turning cycles, and $c = |\mathcal{L}| + |\mathcal{R}|$.

If each edge appears exactly once in $\mathcal{L}$ and exactly once in $\mathcal{R}$ and if $c = n+2$ then the peer code is realizable. We may construct a cellular 2-sphere from discs whose boundaries correspond to the turning cycles of $\mathcal{L}$ and $\mathcal{R}$, and whose 1-skeleton is an immersion that yields our given peer code.

Since we may enumerate permuations of n elements, and may designate crossings as Type I or Type II in only a finite number of ways, and there are only a finite number of ways that we may allocate commas to denote link components, we may determine how many realizable peer codes are possible with n crossings and m components. This has been done by computer search to produce the following table.

number of	realizable peer codes		
crossings, n	$m=1$	$m=2$	$m=3$
3	2	0	0
4	4	8	48
5	12	112	144
6	84	468	1120
7	394	2736	10800
8	1972	17416	68304
9	10604	101696	487296
10	56420	620656	
11	309124		

Clearly, these realizable codes contain many symmetries and redendancies. Reversing the crossing types results in a reflection of the immersion. Starting the numbering of a component at a different edge, possibly reversing the orientation at the same time, or numbering the components in a different order, produces a different code for the same diagram. The above table was calculated assuming no Reidemeister type I moves, another redundancy that may be detected by a computer is when a diagram is a connected sum. Removing these symmetries and redundancies results in the following table.

number of	realizable peer codes			
crossings, n	$m = 1$	$m = 2$	$m = 3$	$m = 4$
3	1	0	0	0
4	1	1	0	0
5	2	1	0	0
6	3	4	2	0
7	10	7	1	0
8	27	27	7	1
9	101	77	19	1
10	364	341		
11	1610			