

## A METHOD OF ENCODING GENERALIZED LINK DIAGRAMS

CHAD MUSICK

*Graduate School of Mathematics, Nagoya University,  
 Furocho, Chikusa-ku, Nagoya 464-8602, Japan  
 ce.musick@yahoo.com*

Received 2 August 2012

Accepted 4 April 2013

Published 14 May 2013

### ABSTRACT

We describe a method of encoding various types of link diagrams, including those with classical, flat, rigid, welded, and virtual crossings. We show that this method may be used to encode link diagrams, up to equivalence, in a notation whose length is a cubic function of the number of “riser marks”. For classical knots, the minimal number of such marks is twice the bridge index, and a classical knot diagram in minimal bridge form with bridge index  $b$  may be encoded in space  $\mathcal{O}(b^2)$ . A set of moves on the notation is defined. As a demonstration of the utility of the notation we give another proof that the Kishino virtual knot is non-classical.

*Keywords:* Knot representation; virtual knots; welded knots; flat knots; diagram encoding; Kishino; generalized link diagrams.

Mathematics Subject Classification 2010: 57M25

### 1. Introduction

A link  $L$  may be viewed as an immersion of circles into some space  $M$ , the exact  $M$  depending on the type of the link. Typically, we are interested in equivalence classes of links given by ambient isotopies in  $M$ ; however, we often categorize these equivalence classes by looking at planar projections and defining equivalence by referring to a set of diagrams reachable from any good projection by a set of local changes.

In this paper, we strive to achieve a balance between the global nature of ambient isotopies in space and the localized nature of diagrammatic moves. In order to do this for a specific link, we adopt a method that relies on a combinatorial sentence and looks at moves on partial diagrams arising from this sentence. A method is given for forming a combinatorial sentence from a diagram, and a corresponding method is given for drawing a diagram from the combinatorial sentence. This gives both a method of encoding links and a method of making moves on links. We also

show how the new notation may be given in a fixed-length form. This is particularly helpful for those diagrams that have a large number of crossings or points around which the winding number of the curve is high.

Two applications of this notation and the associated moves are given. We provide an elementary proof that the Kishino virtual knot is non-classical, and we give a 3-bridge presentation of the 11-crossing knot  $k11a1$ .

## 2. Standard Construction

We wish to demonstrate a method of encoding link diagrams generally. Historically, link diagrams referred to a projection to  $S^2$  or  $\mathbb{R}^2$  of a union of disjoint circles embedded in  $S^3$  or  $\mathbb{R}^3$ . However, our primary interest is in the diagrams as objects in their own right. An overview of the subject is given in Nelson [8]. We therefore make the following definition.

**Definition 2.1 (Link diagram).** A *link diagram* is a planar graph  $G$  in  $S^2$  that meets the following criteria:

- (1) each vertex is 4-valent, and
- (2) in a neighborhood around each vertex, there is a pattern (in, in, out, out) to the orientation of edges, and
- (3) each of the vertices has been assigned one of the set of crossing types {classical +, classical −, flat, virtual, welded (up), welded (down), rigid}.

### 2.1. Crossing types

We handle six distinct crossing types, each of which has its own diagrammatic rules. These crossing types are classical, flat, rigid, welded (both up and down varieties), and virtual. Figure 1 shows the graphical notation we use for each type. All strands are oriented, but orientations are only shown where they are relevant.

### 2.2. Allowed moves

Figure 2 shows the possible local changes. Which changes are allowed depends on the crossing type of the labeled crossings. Our general rule is to take all moves that involve crossing types possible in the diagrams, but we will also give an explicit

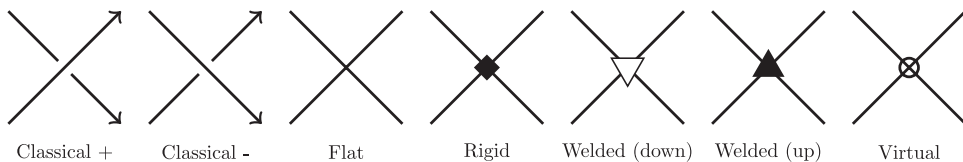


Fig. 1. The different types of crossings.

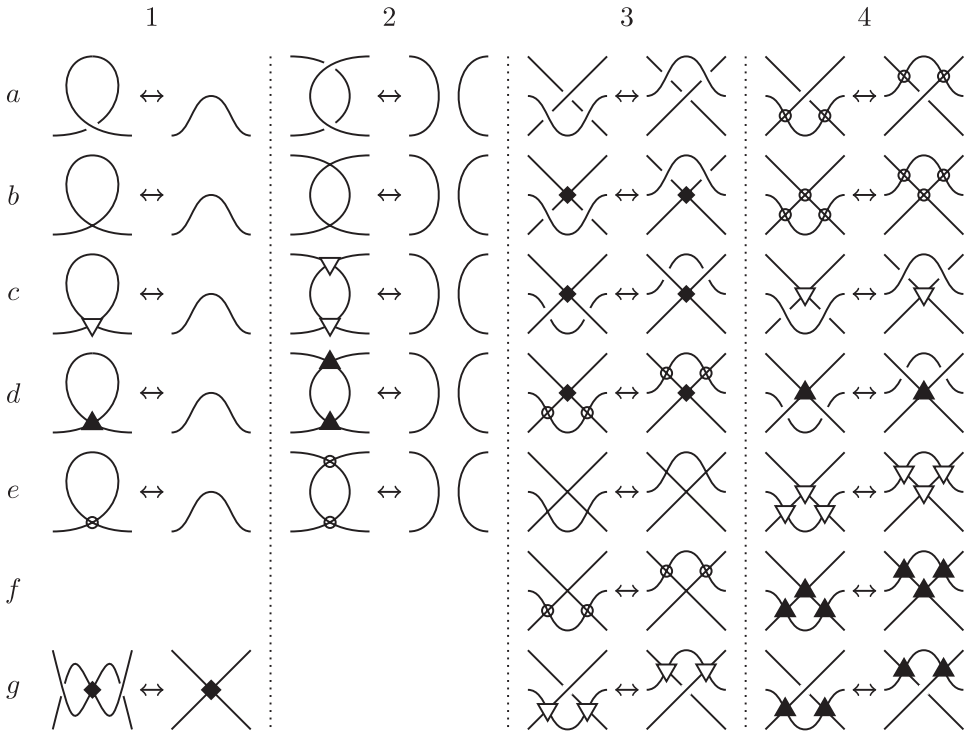


Fig. 2. Local moves.

list for each diagram type. We will refer to moves by the letter and number shown in Fig. 2.

For classical links, local changes to crossings are given by the Reidemeister moves, shown as moves  $a1$ ,  $a2$ , and  $a3$ . Details may be found in standard textbooks, such as Kauffman [5] or Murasugi [7].

For virtual links, which contain classical and virtual crossings, we refer to [4]. Moves permitted for virtual links are  $a1$ ,  $e1$ ,  $a2$ ,  $e2$ ,  $a3$ ,  $a4$ , and  $b4$ . For virtual links that contain rigid crossings, we add moves  $b3$ ,  $c3$ ,  $d3$ , and  $g1$ . For flat virtual links (also described in [4]), which contain flat and virtual crossings, we permit moves  $b1$ ,  $e1$ ,  $b2$ ,  $e2$ ,  $e3$ ,  $f3$ , and  $b4$ .

For welded links, which contain classical and welded (down) crossings, we refer to [2]. Moves permitted for welded links are  $a1$ ,  $c1$ ,  $a2$ ,  $c2$ ,  $a3$ ,  $g3$ ,  $c4$ , and  $e4$ . Welded knots differ from virtual knots by allowing an extra move —  $c4$  — in which ordinary crossings are passed over a welded crossing. We create a distinction between welded (down) knots and the variety given by allowing ordinary crossings to pass under a welded (up) crossing. This alternate theory allows moves  $a1$ ,  $d1$ ,  $a2$ ,  $d2$ ,  $a3$ ,  $g4$ ,  $d4$ , and  $f4$ .

### 3. An Alternate Construction

We now give an alternate construction. We will show that this construction is equivalent, up to allowed moves, to the construction previously exhibited. We first give a formal specification of a language. This specification is given in EBNF, extended Backus–Naur form, defined in the ISO standard 14977:1996 [3]. In EBNF, characters have the following meanings: the vertical bar `|` functions as an “or” indicator, separating the elements of a set of which any one may be chosen; the characters `{ }` enclose items repeated 0 or more times; literals are enclosed in quotes; commas indicate concatenation; the `*` character indicates repetition; the `-` character indicates the set minus operation; and parentheses act as grouping.

**Definition 3.1** (*Tar sentence, virtual word, welded word, basic word, flat word set, rigid word set, circle word, label*). *Tar sentences* and the associated pieces *virtual word*, *welded word*, *basic word*, *flat word set*, *rigid word set*, *circle word*, and *label* are in the space of words defined by the syntax in Fig. 3. Those sentences that match this syntax and can be drawn as described in the next subsection are tar sentences.

Let us give an example. Figure 4 may be represented by the tar sentence `[+3.4.](+5.1.)+6.1.5.+2.3.6.{-1.3.-4.2.};1.2.3.4.5.6.` as demonstrated in future sections; here, we are interested in the component pieces of the syntax. The basic letters of the language are integers — which we separate by the use of the `.` character — and the punctuation marks `{ }` `( )` `+` `-` `;` and `!`. In the tar sentence we have chosen, our letters are the integers 1–6 and a subset of the punctuation. The basic words are `+3.4.`, `+5.1.`, `+6.1.5.`, `+2.3.6.`, `-1.3.`, and `-4.2.`. Additionally, we have the circle word `;1.2.3.4.5.6.`

We note that the number of basic words is equal to the number of integers used. This will be true of all tar sentences because each basic word (in conjunction with the circle word of the sentence) describes, up to an end-point fixing homotopy, an embedding of a path into a 2-sphere with holes. As well, the set of integers that

```
tar sentence = {(virtual word | welded word)}, {(basic word |
    flat word set | rigid word set)}, {welded word}, circle word;
virtual word = "[" , basic word , "]";
welded word = "(" , basic word , ")";
basic word = ("+" | "-") , 2 * label , {label};
flat word set = "{" , basic word , {basic word} , "}";
rigid word set = "!" , 2 * basic word;
circle word = ";" , 2 * label , {label};
label = digit - "0" , {digit} , ".";
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9";
```

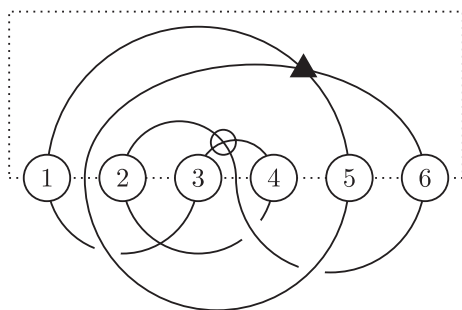
Fig. 3. Formal syntax for tar sentences.

are at the start of a basic word will be the complete set of integers used; the same is true for integers that are at the end of a basic word. No basic word will begin and end with the same integer. The first two basic words are enclosed like this:  $[+3.4.]$  and  $(+5.1.)$ . This indicates that the two basic words are of virtual and welded *types*, respectively.

The rules for turning word types into crossing types are given in the next subsection, but determining the type of the word can be done directly from the syntax: virtual words are enclosed in  $[]$ ; welded words are enclosed in  $()$  — whether it is welded up or down depends on the placement of the welded word in the sentence; all of the basic words in the same flat hierarchy are enclosed in  $\{\}$ ; rigid words (which always come in pairs) are preceded by a  $!$ ; and every basic word with no other type is classical. The shape of the path described by a word depends only on the basic word. Looking at Fig. 4 again, we see that the last two basic words are enclosed in braces; the crossing between the pieces of the diagram represented by these two words — the arc from 1 to 3 and the arc from 4 to 2 is flat. The only type of crossing that has a notion of upper and lower is the classical crossing. For these, which is upper and which is lower is determined by the order of the words: upper words precede lower words.

The circle word is a permutation of the integers used to form the basic words; it may be identified in the tar sentence because it begins with a  $;$  character and is always the final word of the sentence. We see that in the diagram the labels are listed in the circle word in the same order that they are encountered when traveling around the dotted line. We will see how this gives shape to a diagram and how it may be chosen from a diagram in future subsections. In essence, the circle word gives the location of the holes in the 2-spheres used by the basic words.

Of special note is that a link diagram may be represented by a multitude of distinct tar sentences. As a result, many of the changes listed as moves for the tar sentences may not alter the diagram.



$[+3.4.](+5.1.)+6.1.5.+2.3.6.\{-1.3.-4.2.\};1.2.3.4.5.6.$

Fig. 4. A tar sentence with various crossing types and an associated diagram; the link orientation is from 1 toward 3.

### 3.1. How to draw a diagram from a tar sentence

A link diagram in  $S^2$  may be constructed from a tar sentence  $L$  by the following steps. The crossing types of a diagram are determined by the rules given after the construction process.

- (1) Let  $V_n = v_1, v_2, \dots, v_n$  be the sequence of labels in the circle word of  $L$ .
- (2) Let  $c(v_i)$  be the number of occurrences of the label  $v_i$  in the basic words of  $L$ .
- (3) If the elements of  $V$  are not all distinct or the number of distinct labels in the basic words of  $L$  is not  $n$ , then  $L$  is not valid.
- (4) Let  $m_1 = m_{n+1} = 0, m_j = \sum_{i=1}^{j-1} c(v_i) - 1$ .
- (5) Let  $k = m_n + c(v_n) - 1$ .
- (6) Draw the labels  $v_1, v_2, \dots, v_n$  at angles  $\frac{2\pi m_1}{k}, \frac{2\pi m_2}{k}, \dots, \frac{2\pi m_n}{k}$ , respectively, on the unit circle of  $\mathbb{R}^2$ .
- (7) For each basic word  $w$  of  $L$ , we construct a curve on  $S^2 \in S^3$  in the following manner.
  - (a) If  $w$  begins with  $+$ , the first arc goes through the upper hemisphere of  $S^2 \in S^3$ . Otherwise, the first arc goes through the lower hemisphere of  $S^2 \in S^3$ .
  - (b) Arcs alternate between hemispheres and project (by  $(x, y, z) \mapsto (x, y)$ ) to  $\mathbb{R}^2$  as chords of the unit circle.
  - (c) The first arc begins at the label matching the first label of  $w$ ; the last arc ends at the label matching the last label of  $w$ .
  - (d) Each pair of consecutive labels  $v_i v_j$  in  $w$  projects to a chord with ends in the arcs  $\left[\frac{2\pi m_i}{k}, \frac{2\pi m_{i+1}}{k}\right)$  and  $\left[\frac{2\pi m_j}{k}, \frac{2\pi m_{j+1}}{k}\right)$  of the unit circle.
- (8) The ends of the chords of the previous step are chosen to be evenly spaced around the unit circle and to minimize the number of crossings.
- (9) If any basic word cannot be drawn to avoid self-crossings of the curve,  $L$  is not valid.

Figure 5 shows some basic words with projections of their diagrams. It is interesting to notice, assuming a circle word containing  $1.2.3.4.$ , that  $+1.2.4.$  and  $+1.2.4.2.4.$  are valid basic words but  $+1.2.4.2.$  is not.

The following rules determine the type of every crossing between two curves drawn from words  $w_i$  and  $w_j$  in  $L = \dots w_i \dots w_j \dots$ :

- (1) if  $w_i$  or  $w_j$  is a virtual word, then the crossings are virtual; otherwise,
- (2) if  $w_i$  or  $w_j$  is a welded word and the other is not or there is no word  $w_k$  between  $w_i$  and  $w_j$  with  $w_k$  non-welded and non-virtual, then the crossings are welded — up for a welded word at the beginning of the sentence, down for a welded word in the group adjacent to the circle word; otherwise,
- (3) if  $w_i$  and  $w_j$  are welded, then the link is non-standard and the crossings are also non-standard; otherwise,
- (4) if  $w_i$  and  $w_j$  are in the same rigid word set, then the crossing is rigid; otherwise,

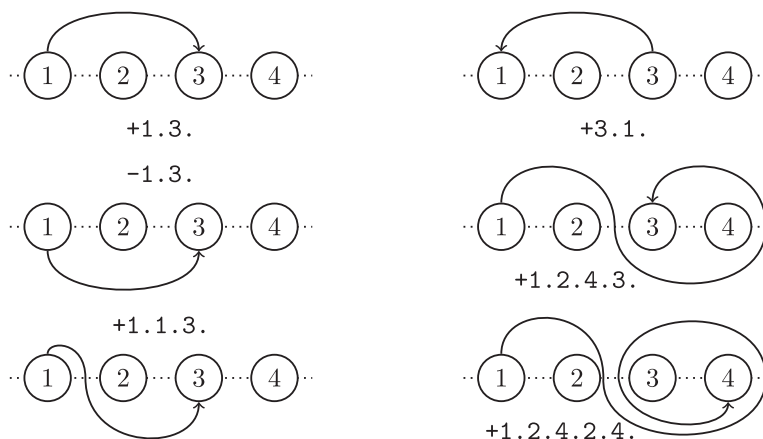


Fig. 5. Some basic words and associated diagrams. All have a circle word containing  $1.2.3.4..$

- (5) if  $w_i$  and  $w_j$  are in the same flat word set, then the crossings are flat; otherwise,
- (6) the crossing is classical, and  $w_i$  crosses over  $w_j$ .

We will give an interpretation of a crossing between welded (up) and welded (down) trails in the section on moves of a tar sentence.

### 3.2. How to construct a tar sentence from a link diagram

Let  $L$  be a link diagram. To  $L$  we add some nodes by subdividing arcs of  $L$ . We assign each new node, all of which are of degree 2, the type *riser mark*. Let  $L'$  be a link diagram created in this way by subdividing each arc of  $L$  twice.

We now consider paths of length 2 where the center node is of degree 4 and the path forms a transverse crossing with the path comprising the other two paths of the center node. To each of these paths we assign the same type as the type of the center node. This leaves paths of length 1 whose ends are both nodes of degree 2. To each of these paths we assign the classical type.

**Definition 3.2 (Fully partitioned link diagram).** A projection to  $\mathbb{R}^2$  of a link diagram in which all arcs have been twice subdivided and types have been assigned such that every arc is assigned a type and no arc has a type different than its head or tail node, whichever is a crossing, is a *fully partitioned link diagram*.

**Definition 3.3 (Riser mark, trail).** A *riser mark* is a degree two vertex. A *trail* is a path that begins and ends at a riser mark and crosses each degree 4-vertex transversely.

**Definition 3.4 (Binding circle, inside, outside).** A *binding circle* of a fully partitioned link diagram is an oriented, self-avoiding loop which contains each riser

mark of the diagram and crosses all other points of the diagram transversely. The region bordering the binding circle is *outside* the binding circle, and the finite region bounded by the binding circle is *inside* the binding circle. The binding circle is oriented such that the winding number of any point inside the binding circle is one.

We label the riser marks of  $L$  with distinct, positive integers and construct the basic words of  $L$  in the following way:

- (1) If there is a neighborhood of the tail of the trail distinct from the outside of the binding circle, then the first letter is  $+$ .
- (2) If there is a neighborhood of the tail of the trail distinct from the inside of the binding circle, then the first letter is  $-$ .
- (3) The second and third letters are the label of the riser mark at the tail of the trail and the  $.$  character, respectively.
- (4) At each crossing of the trail and the binding circle, taking crossings sequentially along the trail, the label of the riser mark first encountered when following the binding circle counter to its orientation from the crossing is appended, followed by the  $.$  character.
- (5) The last two letters of the basic word are the label of the riser mark at the head of the trail and the  $.$  character.

From the basic word  $w$  of each trail, we construct the tar sentence in the following order.

- (1) The virtual trails are written in any order by writing  $[w]$  for each.
- (2) The welded (up) trails are written in any order by writing  $(w)$  for each.
- (3) The classical, rigid, and flat trails are written in an order determined by their classical crossings. If  $a$  crosses over  $b$ ,  $a$  must appear before  $b$ .
- (4) Flat trails  $w_1$  and  $w_2$  that share a crossing are written  $\{w_1 w_2\}$ .
- (5) Rigid trails  $w_1$  and  $w_2$  that share a crossing are written  $! w_1 w_2$ .
- (6) The welded (down) trails are written in any order by writing  $(w)$  for each.
- (7) The binding circle is written as  $; l_1 . l_2 . \dots$  where  $l_1$  is the label of an arbitrarily chosen riser mark and each  $l_i$  is the label of the next riser mark encountered when following the binding circle in its orientation.

**Proposition 3.5.** *The tar sentence of a link diagram created as described allows reconstruction of the diagram up to isotopy.*

**Proof.** A link diagram with the riser marks and binding circle described is partitioned into disjoint crossings and arcs by the removal of the binding circle. The tar sentence describes a map for stitching together the pieces of the diagram.  $\square$



#### 4. Moves on the Alternate Construction

The construction of a tar sentence given produces a representation of a link that is, in most cases, longer than a representation that relies on enumerating the same set of crossings. We proceed now to show two reasons why it is of interest. First, the moves on tar sentences are more global in nature than the local crossing changes of the standard representations. Second, the tar sentence may be compacted to a form whose length depends only on the number of riser marks. Because the minimal number of riser marks is twice the bridge index for classical links and comparable for many non-classical links (specifically excluding rigid vertices), links with very high crossing numbers can often be expressed in a brief way. In this section, we describe the moves on tar sentences.

Given a tar sentence, we can make moves of the following types:

- (1) riser removal (Definition 4.1),
- (2) trail swap (Definition 4.2),
- (3) duplicate removal (Definition 4.3),
- (4) riser addition (Definition 4.4),
- (5) trail type conversion (Definition 4.5),
- (6) riser move (Definition 4.10),
- (7) trail move (Definition 4.11),
- (8) rigid flip (Definition 4.12).

**Definition 4.1 (Riser removal).** A *riser removal* of a sentence is the concatenation of two basic words. In order to be concatenated, the two basic words must meet the following requirements:

- (1) neither of the words may be in a rigid word set,
- (2) when the punctuation, excepting  $+$   $-$ , is removed, the basic words must be consecutive,
- (3) the final label of the first word must be the first label of the second,
- (4) the trails represented by the basic words must not cross,
- (5) if either of the words is in a flat word set, the other word must not intersect with any trails represented by members of that set,
- (6) if the words are of different types, then each must be either basic or flat,
- (7) there are at least three basic words in the link component of the diagram of which the words are a partial representation.

For basic words  $a$  and  $b$  which meet these criteria, the concatenation is the result of the following process.

- (1) If the number of labels in the first word is even, then:
  - (a) if the first letters of  $a$  and  $b$  are the same, then remove the final label of  $a$  and the first letter and first label of  $b$ ;

- (b) if the first letters of  $a$  and  $b$  differ, then remove the final label of  $a$  and the first letter of  $b$ .
- (2) If the number of labels in the first word is odd, then:
  - (a) if the first letters of  $a$  and  $b$  are the same, then remove the final label of  $a$  and the first letter of  $b$ ;
  - (b) if the first letters of  $a$  and  $b$  differ, then remove the final label of  $a$  and the first letter and first label of  $b$ .
- (3) In all basic words replace all occurrences of the last label  $l$  of  $a$  with the label immediately before  $l$  in the circle word of the sentence; if  $l$  is the first label of the circle word, replace  $l$  with the last label of the circle word.
- (4) Replace the punctuation indicating word type (e.g., virtual word, flat word set, etc.) according to the following rules:
  - (a) words of the same type keep that type;
  - (b) a basic word and a flat word form a word that belongs in the same flat word set as the original flat word;
  - (c) if both words are flat and from different flat word sets, the two sets are combined.

Figure 6 shows a partial diagram with riser removal.

**Definition 4.2 (Trail swap).** A *trail swap* is the exchange of the position of two consecutive trail elements of a sentence. Trail elements are virtual words, welded words, rigid word sets, flat word sets, and basic words not in a virtual word or welded word. This exchange is permitted in the following cases. In all other cases, exchange is forbidden:

- (1) both are members of the same flat word set,
- (2) both are basic words and the trails represented by the two do not cross,
- (3) both are virtual,
- (4) both are welded,
- (5) one is virtual and the other is welded,
- (6) each is either a basic word, flat word set, or rigid word set and there are no crossings between trails represented by the first and the second.

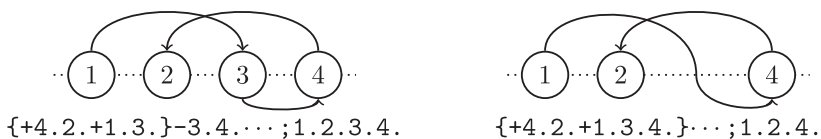


Fig. 6. A riser removal.

A trail swap has induces no diagrammatic change. As an example, the left-hand side of Fig. 6 may be rewritten as  $\{+1.3.+4.2.\}-3.4.\dots;1.2.3.4.$  without altering the diagram.

**Definition 4.3 (Duplicate removal).** A *duplicate removal* is the replacement of a pair of adjacent, identical labels according to the following rules:

- (1) if the duplication occurs in a rigid word set, then no change occurs; otherwise,
- (2) if the letter before the duplication is a  $+$ , then one of the duplicates is removed and the  $+$  is replaced by a  $-$ ; otherwise,
- (3) if the letter before the duplication is a  $-$ , then one of the duplicates is removed and the  $+$  is replaced by a  $+$ ; otherwise,
- (4) if the repeated labels are the final two labels in a word, then one of them is removed; otherwise,
- (5) if none of the previous cases apply, then both of the duplicates are removed.

Figure 7 shows a duplicate removal. The change from  $+1.1.3.$  to  $-1.3.$  in Fig. 5 is also an example of a duplicate removal.

**Definition 4.4 (Riser addition).** A *riser addition* is the replacement of a basic word, not part of a rigid word set, by a pair of basic words. It occurs by the following procedure.

- (1) We choose a basic word.
- (2) We let the final label of the word be  $l$ .
- (3) A new label, distinct from any label in the sentence, is chosen. We call this new label  $t$ .
- (4) In the circle word, we replace  $l$  by the sequence  $tl$ .
- (5) We replace the final label of the basic word by  $t$ . We insert the word  $+lt$  just after the chosen word with the same word type as the chosen word. If the chosen word is flat, the new word is placed in the same flat word set.

Figure 6 shows a riser addition if the diagram is read from right to left.

**Definition 4.5 (Trail type conversion).** A *trail type conversion* is the replacement of one trail element by a different trail element containing the same basic words but of a different type. This may be done according to the following rules.



Fig. 7. A duplicate removal.

- (1) A virtual (welded) word for which the trail represented by the word has no crossings with any non-virtual (non-welded) trails and that follows all other virtual words and welded words representing welded (up) trails may be replaced by a basic word.
- (2) A flat word set containing only one basic word may be replaced by that basic word.
- (3) A basic word representing a flat trail that has no crossings with any of the trails represented by other members of its flat word set may be moved to immediately precede the flat word set. If this leaves an empty set, the previous rule is used instead.
- (4) A basic word with no classical crossings and which is preceded only by virtual and welded words may be replaced by a virtual or welded word.
- (5) A basic word with no classical crossings and which is followed only by welded words may be replaced by a welded word.

The left-hand side of Fig. 6 may be written as  $\{+4.2.+1.3.-3.4.\} \cdots; 1.2.3.4.$  instead of  $\{+4.2.+1.3.\}-3.4. \cdots; 1.2.3.4..$  This is a trail type conversion, and like all such conversions, does not affect the diagram.

For the remaining two moves, we need to define what we mean by a *level diagram* and isotopies of these diagrams.

**Definition 4.6 (Level diagram, standard level diagram set).** A *level diagram* of a tar sentence is a partial reconstruction of a link diagram from the tar sentence by the following method.

- (1) A trail element of the sentence or a basic word of a flat word set is chosen.
- (2) The binding circle is drawn and labeled according to the circle word.
- (3) Each basic word in the element is drawn.
- (4) Labels are removed from the binding circle according to the rules:
  - (a) If the trail element is virtual or welded, all labels except those at the ends of the trail are removed; otherwise,
  - (b) a label is removed if it does not belong to a virtual word and the basic words that contain this label as their first or last label both precede or follow the chosen element.

We let the *standard level diagram set* be the set of level diagrams given by taking the level set of each trail element except for flat word sets and taking the level set diagrams of each basic word in a flat word set.

Figures 11 and 12 show a link diagram and the corresponding level diagrams.

**Definition 4.7 (Ambient isotopy).** An *ambient isotopy* is a continuous mapping  $f : S^2 \times [0, 1] \rightarrow S^2$  such that every  $f_t : S^2 \rightarrow S^2$  given by  $f_t(X) = f(X, t)$  is a homeomorphism of  $S^2$ .

**Definition 4.8 (Level isotopy).** A *level isotopy* is an isotopy of a level diagram  $L$  that is determined by an ambient isotopy  $f$  of  $S^2$  such that  $f(L, 0) = L$ .

**Definition 4.9 (Riser isotopy).** A *riser isotopy* of a tar sentence is a set of level isotopies  $\{f^1, f^2, \dots, f^n\}$ , one for each corresponding level diagram  $d_1, d_2, \dots, d_n$  in the standard level diagram set, such that all riser marks except one, which we assume is at position  $r$ , are constant under  $f^i$  if they are drawn in  $d_i$  and for every  $i, j$  the equality  $f^i(r, t) = f^j(r, t)$  for all  $t \in [0, 1]$  whenever  $d_i$  and  $d_j$  both draw the riser label at  $r$ .

**Definition 4.10 (Riser move).** A *riser move* is the replacement of all basic words by the basic words of the diagram that results from choosing a riser isotopy  $f^1, f^2, \dots, f^n$  on levels  $d_1, d_2, \dots, d^n$  and replacing each  $f^i(d_i, 0)$  by  $f^i(d_i, 1)$ .

Figure 8 shows part of a riser move. The trail represented by the basic word  $+1.3.$  is altered to the trail represented by  $+1.2.3.$  to accommodate the new location of riser mark 2, which has moved in the manner shown by the dashed line. The circle word must, likewise, be modified to contain  $1.3.2.4.$  instead of  $1.2.3.4..$

**Definition 4.11 (Trail move).** A *trail move* is the replacement of a basic word by the word describing the result of a level isotopy that keeps all labels appearing on the level diagram constant.

**Definition 4.12 (Rigid flip).** A *rigid flip* is the replacement of a rigid word set  $!w_1w_2$  by the rigid word set  $!w'_2w'_1$  where  $w'_1$  and  $w'_2$  are the result of changing all  $+$  to  $-$  and all  $-$  to  $+$  in  $w_1$  and  $w_2$ , respectively. A rigid flip is allowed if and only if the sentences obtained by replacing  $!w_1w_2$  with  $w_1w_2$  and with  $w'_2w'_1$  are equivalent.

**Theorem 4.13.** For tar sentences  $L_1$  and  $L_2$ , there is a sequence of tar sentence moves that transforms  $L_1$  into  $L_2$  if and only if the links represented by  $L_1$  and  $L_2$  are equivalent.

**Proof.** We proceed by demonstrating that every move of Fig. 2 can be performed by tar sentence moves and that none of the forbidden moves of Fig. 9 can be performed by tar sentence moves.

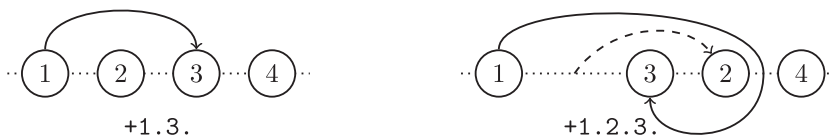


Fig. 8. A riser move by riser 2.

For moves  $a1$ ,  $b1$ ,  $c1$ ,  $d1$  and  $e1$ , we see that the right-hand side may be a basic word  $+1.2.$  with  $1.2.$  appearing in the circle word by a suitable choice of labels, riser additions, and riser removals. By trail swaps, this may be at whatever position in the sentence is necessary for the appropriate trail type conversion. The left-hand side may then be written as  $+1.3.-3.4.+4.2.$  with  $1.3.4.2.$  appearing in the circle word by two riser additions and a riser move. To move in the reverse direction, we reverse these moves, performing a riser move and then two riser removals.

For moves  $a2$ ,  $b2$ ,  $c2$ ,  $d2$  and  $e2$ , we assume that the arcs on the right-hand side are each a basic word. If not, we may perform riser additions and removals to make them so. By a riser addition and a riser move, we can create the left-hand side. To move from the right-hand side to the left-hand side, we note that riser additions and removals will suffice because the number of crossings is, by definition, minimal. If each of the arcs on the right-hand side has no interior riser marks, then a minimal drawing of them will have no crossings.

For the moves of columns 3 and 4, we note that welded and virtual words may be replaced freely by alternate words because their level diagrams consist solely of their endpoints and trail and so they are isotopic to every path from their start to their end. This handles all of the cases involving 2 or more virtual or welded crossings. We note that the remaining moves may be constructed in this same way because the trails can be cut sufficiently to allow this freedom for all non-virtual and non-welded crossings. Figure 9 shows moves that are forbidden, and it is not obvious that these cannot be constructed.

To see that the forbidden moves cannot be constructed, we notice that because the central crossings are virtual or welded, then one of the trails involved must either be virtual or welded. As such, a riser mark must appear in the level diagram of the classical trail. This acts as a puncture to prevent any trail move that would be necessary to construct this move. If such a move is going to be constructed, it cannot be locally, and this is sufficient prevention.

The last move to construct is  $g1$ . Figure 10 shows how to do this by three riser moves and a rigid flip.  $\square$

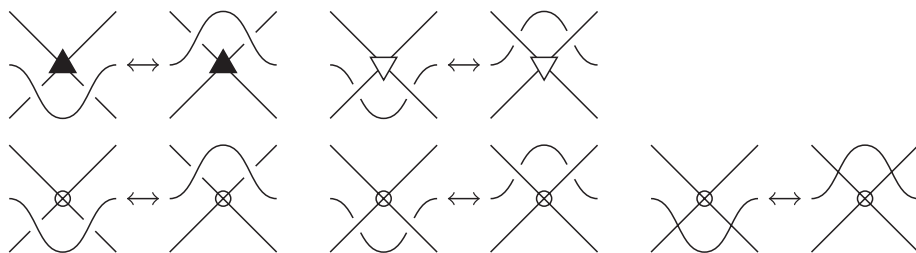


Fig. 9. Forbidden local moves.

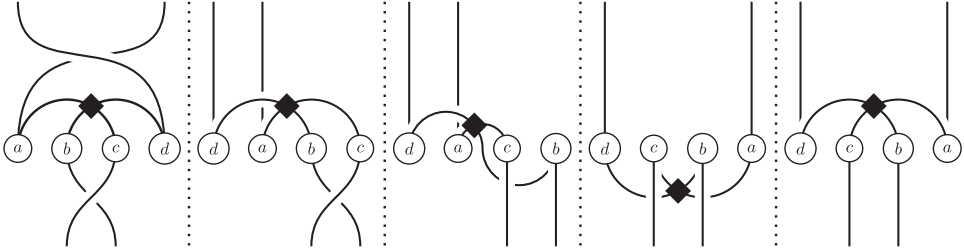


Fig. 10. Constructing the rigid vertex flip.

#### 4.1. A representation whose size depends only on the number of words

It may be that the tar sentence of a particular diagram is long but that the number of words in the sentence is low. In this case, it would be convenient to have a representation that exploits this fact. We show such a representation here. To achieve this representation we take the following steps.

- (1) Rename the labels so that the circle word is ;1.2.3. ...n.. We regard the labels as having the corresponding numeric values.
- (2) Perform duplicate removals repeatedly until no more duplicate removals are possible.
- (3) Replace the first label,  $l$ , of each basic word with  $* l$ .
- (4) Replace the last label,  $l$ , of each basic word with  $* l$ .
- (5) We now regard the character  $*$  as part of the label that follows it.
- (6) Place a , between each pair of adjacent basic words.
- (7) Let  $a, b$  be arbitrary labels. Replace each subword  $+ab$  with  $ab|-b$  if  $a < b$  and otherwise with  $ba|-b$ . Similarly, replace each subword  $-ab$  with  $ab|+b$  if  $b < a$  and otherwise with  $ba|+b$ .
- (8) Shorten any phrase  $|+aw$  or  $|-aw$  to  $w$  where  $a$  is a label and  $w$  is one of , { } ( ) ! ; .
- (9) Repeat the previous two steps until no further changes have been made. The arcs of the transformed trails are now separated by | characters, and the transformed trails are separated by other punctuation.
- (10) In a sequence of  $n$  square matrices with all entries 0 and dimension  $2n$ , modify the matrices as follows.
  - (a) For every pair of consecutive labels  $i, j$  let  $r = 2i$ . If  $i$  contains a  $*$ , subtract 1 from  $r$ . Let  $c$  be the analogous number for  $j$ .
  - (b) If  $ij$  arose from the transformation of the  $k$ th basic word, add 1 to the  $r$ th row and  $c$ th column of the  $k$ th matrix.
- (11) Let the entry  $M_{1,1}$  of each matrix be determined by its type, with 0 classical, 1 virtual, 2 welded, 3 rigid, and higher numbers used for flat crossing hierarchies.

- (12) The sequence of matrices is a representation of the same link as the tar sentence from which it was constructed.

**Proposition 4.14.** *The sequence of matrices given can be used to construct a tar sentence, up to orientation changes of the components.*

**Proof.** There are two key facts. One is that trails are self-avoiding. From this, we may reconstruct the basic words by drawing the arcs in the hemispheres of  $S^2$  disjointly and then connecting the ends. Each basic word is represented by one matrix. For each matrix, we draw the curve by dividing the circle into  $n$  pieces. We label the pieces with odd numbers  $1, 3, \dots, 2n - 1$  and the interstices with even numbers  $2, 4, \dots, 2n$ . Entries in the upper triangle of the matrix represent arcs in the upper hemisphere; entries in the lower triangle of the matrix represent arcs in the lower hemisphere. In either case, the arcs are drawn so that they project as chords from the region indicated by the row to the region indicated by the column. From this curve, either the basic word or its reverse may be read.

We recover the order of the basic words from the order of the matrices. We recover the type of the word, and any flat set hierarchy, from the upper left entry. We recover the basic word or its reverse by the process just described. The circle word is given by  $;1.2.\dots n..$  We may reverse at most half of the basic words so that the orientation of the link components is consistent. After this, the entire tar sentence, up to orientation change, is recovered.  $\square$

**Remark 4.15.** If  $k$  trails of the same type do not cross, then the sum of their matrices gives back these  $k$  curves upon reconstruction. Using this technique, we may represent classical links in only two matrices — one for bridges, one for underpasses. Suitably transformed diagrams of other types, with the notable exception of those containing flat hierarchies or rigid crossings, may likewise be represented by a small number of matrices.

## 5. Two Examples Where the Notation is Useful

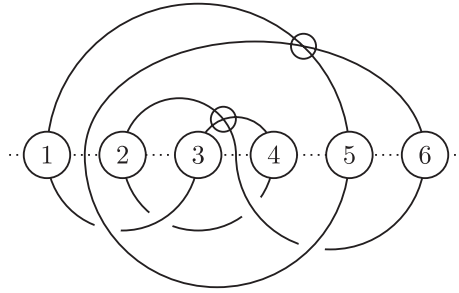
We give two examples of the utility of the notation.

### 5.1. *The sentence-style notation: The Kishino virtual knot is non-classical*

In the first we use the sentence style of the notation. We examine the Kishino virtual knot of Kishino and Saito [6] and provide another proof that it is non-trivial and non-classical by use of the tar sentence and its moves. Figure 11 shows a diagram of the knot and a corresponding tar sentence.

**Proposition 5.1.** *The Kishino virtual knot is non-classical.*





As virtual:  $[+3.4.] [+5.1.] +6.1.5. +2.3.6. -1.3. -4.2. ; 1.2.3.4.5.6.$   
 As flat virtual:  $[+3.4.] [+5.1.] \{+6.1.5. +2.3.6. -1.3. -4.2.\}; 1.2.3.4.5.6.$

Fig. 11. A tar sentence of the Kishino virtual knot.

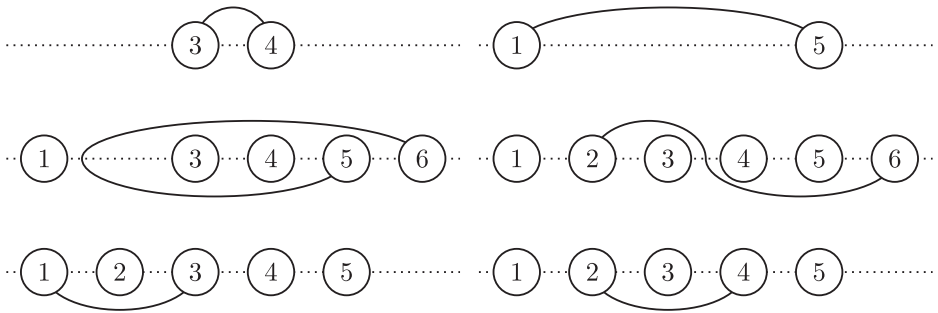


Fig. 12. Level diagrams of the Kishino virtual knot.

**Proof.** We proceed by examining the level diagrams and considering moves. The level diagrams are shown in Fig. 12.

We see that neither of the virtual trails may be moved by a trail move to avoid all of its crossings, nor may we place all of the virtual crossings on one trail. This is easily seen by noticing that the trails  $+2.3.6.$  and  $-4.2.$  form a circle around the riser mark 3, and the trails  $+6.1.5.$  and  $+2.3.6.$  form a circle around the riser mark 5. The ends of the virtual trails therefore lie in three distinct regions and cannot be made to lie on one trail because then the ends would only lie in two distinct regions.

Because the ends of the virtual trails appear on every classical level diagram, no riser move can free them from these regions, and no trail move or riser addition can change the separation of the virtual ends. Since there are two virtual crossings and must be at least two virtual trails, the number of virtual words in the sentence cannot be reduced. A link of which every tar sentence has more than 0 virtual words cannot be classical.  $\square$

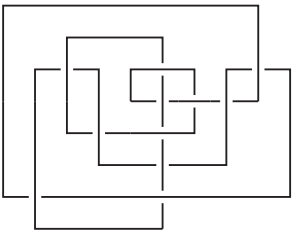


Fig. 13. Knot 11a1 in an 11-crossing alternating projection.

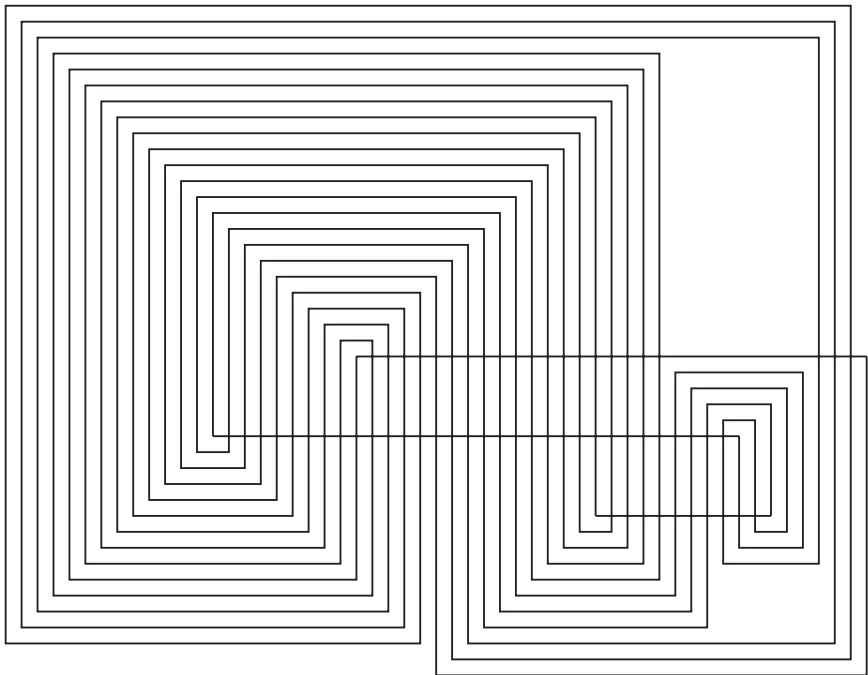


Fig. 14. Knot 11a1 in a 3-bridge projection. At all crossings, the vertical strand is above.

**5.2. The reduced notation style: A 3-bridge diagram of knot 11a1**

Figure 13 shows knot 11a1 (we adopt the numbering scheme of [1]) in a minimal crossing form. Figure 14 shows this same knot in a 3-bridge, 64-crossing form. Because the matrices produced by the reduced notation style representation are sparse, we list only the non-zero entries of these, and we do so in symbolic form. That is, we list the corresponding sentence labels rather than the matrix coordinates. Because the representation is in the form of bridges and underpasses, we sum together the three bridge matrices and also sum together the three underpass matrices.

The result of this is the representation:

$$\begin{aligned} \text{Bridges: } (*1, 5) &= 1, (1, 2) = 7, (1, 4) = 2, (1, *5) = 1, (1, 5) = 13 \\ (2, 4) &= 3, (*3, 4) = 1, (3, 4) = 1, (5, 6) = 8, (3, 2) = 1 \\ (*4, 2) &= 1, (4, 2) = 4, (5, 1) = 14, (5, *2) = 1, (5, 2) = 4 \\ (5, 4) &= 3, (*6, 1) = 1, (6, 1) = 8 \end{aligned}$$

$$\text{Underpasses: } (*2, *3) = 1, (*4, *1) = 1, (*5, *6) = 1.$$

We may read this representation directly from the diagram by labeling the leftmost point on the central underpass 1, continuing clockwise to choose 2 as the leftmost of the lower underpass, 3 as the rightmost of that same underpass, 4 as the rightmost of the center underpass, 5 as the rightmost of the upper underpass, and 6 as the leftmost of the upper underpass. The binding circle is drawn as straight lines, and the upper and lower underpass are pushed slightly out of the interior of the circle.

## Acknowledgments

This paper is the result of discussions with many participants from the 2012 workshop “Invariants in Low-Dimensional Topology and Knot Theory” at the Mathematisches Forschungsinstitut Oberwolfach. The author is grateful to the Institute and the participants. In particular, J. Scott Carter, Heather Dye, Slavik Jablan, Lou Kauffman and Vassily Manturov each made specific suggestions that have been incorporated into this paper and improved its clarity. The anonymous reviewer also made helpful suggestions, and the author is grateful to him or her.

## References

- [1] D. Bar-Natan *et al.*, The knot atlas, <http://katlas.org/>.
- [2] R. Fenn, R. Rimanyi and C. Rourke. The braid permutation group, *Topology* **36**(1) (1997) 123–135.
- [3] ISO/IEC, International Standard ISO/IEC 14977:1996 — Information technology — Syntactic metalanguage — Extended BNF (1996).
- [4] L. H. Kauffman, Virtual knot theory, *European J. Combin.* **20** (1999) 663–690.
- [5] L. H. Kauffman, *Formal Knot Theory* (Dover, 2006).
- [6] T. Kishino and S. Satoh, A note on non-classical virtual knots, *J. Knot Theory Ramifications* **13**(7) (2004) 845–856.
- [7] K. Murasugi, *Knot Theory and Its Applications*, English edn. (Birkhäuser, 1996).
- [8] S. Nelson, The combinatorial revolution in knot theory, *Notices Amer. Math. Soc.* **58**(11) (2011) 1553–1561.