Visual algebraic proofs for unknot detection

Andrew Fish[1], Alexei Lisitsa[2], and Alexei Vernitski[3]

[1] Centre for Secure, Intelligent and Usable Systems, School of Computing,
Engineering & Mathematics, University of Brighton, Brighton UK
andrew.fish@brighton.ac.uk
[2] Department of Computer Science, University of Liverpool, Liverpool, UK
A.Lisitsa@liverpool.ac.uk
[3] Department of Mathematical Sciences, University of Essex, Essex, UK
asvern@essex.ac.uk

Abstract. A knot diagram looks like a two-dimensional drawing of a
knotted rubberband. Proving that a given knot diagram can be untangled
(that is, is a trivial knot, called an unknot) is one of the most famous
problems of knot theory. For a small knot diagram, one can try to find
a sequence of untangling moves explicitly, but for a larger knot diagram
producing such a proof is difficult, and the produced proofs are hard
to inspect and understand. Advanced approaches use algebra, with an
advantage that since the proofs are algebraic, a computer can be used
to produce the proofs, and, therefore, a proof can be produced even
for large knot diagrams. However, such produced proofs are not easy to
read and, for larger diagrams, not likely to be human readable at all.
We propose a new approach combining advantages of these: the proofs
are algebraic and can be produced by a computer, whilst each part of
the proof can be represented as a reasonably small knot-like diagram
(a new representation as a labeled tangle diagram), which can be easily
inspected by a human for the purposes of checking the proof and finding
out interesting facts about the knot diagram.

1 Introduction

A knot diagram looks like a two-dimensional drawing of a knotted rubberband. In
the simplest case, consider a knot diagram (or a rubberband) without crossings
(e.g. see the lower right diagram in Figure 1); this knot diagram is known as the
trivial knot or *unknot*. For this and other basic concepts of knot theory, see any
of the textbooks [1, 12, 9, 10, 13, 16]. A knot diagram which looks knotted may be
really knotted, or it may be then the unknot in disguise, with the diagram (think
rubberband) being able to be untangled by gently pulling some of its parts in
some order, until the diagram does not have any crossings (at which point it is
obvious that it is a trivial knot). An example of a sequence of such untangling
steps is shown in Figure 1; intuitively these correspond to moving a rubberband
in the 3-dimensional Euclidean space without cutting it.

The problem of deciding whether a given knot diagram can be untangled
(that is, is a trivial knot) is one of the most famous problems of knot theory. It
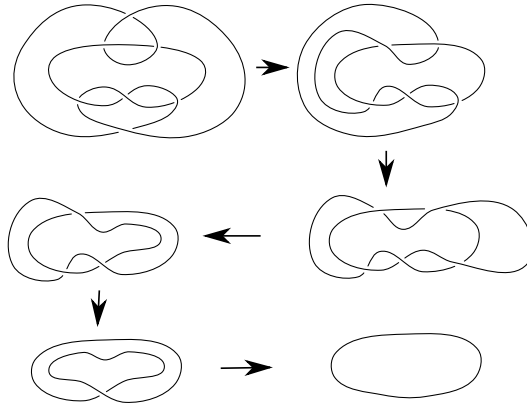
Fig. 1. An example of a step-by-step untangling of a diagram of a trivial knot.

is an interesting problem, having attracted diverse approaches from a number of different areas of mathematics, whilst having an immediate aesthetic appeal.

The immediate, naive approach to the problem is trying to find a sequence of untangling moves explicitly; then the process of untangling can be represented as a sequence of diagrams (as in Figure 1). For small knot diagrams this can be a preferred method, but for larger knot diagrams producing such proofs is difficult [2], and the produced proof becomes hard to inspect (as you can see, even the relatively small proof in Figure 1 has some steps that are not so easy to follow).

A number of more advanced approaches are based on using algebra. Some of these approaches are based on denoting each arc (that is, a continuous unbroken line segment of the diagram) by a letter, as shown in the example in Figure 2, and then proving that all these letters are equal to each other in a certain algebra. Some of the authors' previous research concentrated on such methods of untangling [4, 5, 11], and this paper presents a new twist in this research. (For completeness, note that not all algebraic approaches to untangling are based on labelling arcs; see [3], for example.)
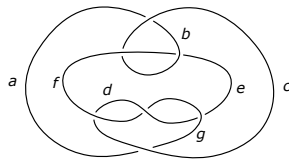
Fig. 2. A labelled knot diagram
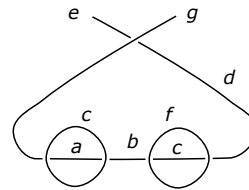
Fig. 3. A labelled tangle diagram

The advantage of such an approach is that since proofs are purely algebraic, a computer can be used to produce the proofs, and, therefore, a proof can be

produced even for a large knot diagram. However, the produced proof, consisting of long chains of abstract equalities, is not easy to read and, for larger diagrams, is not human readable at all. For example, here is a proof showing that the diagram in Figure 2 is a trivial knot (adapted from [19]):

"$bc = ca = gc$, hence, $b = g$; $bf = fc = df$, hence, $b = d$; $gg = gb = gd = eg$, hence, $g = e$; $dg = fd = fb = fg$, hence, $d = f$; $cf = fd = dg = db = dd = df$, hence, $c = d$; $ab = ad = ac = cb$, hence, $a = c$.".

In this paper, we propose a new approach, which combines the advantages of the above approaches. Our proofs are produced algebraically (and can be produced by a computer), as in the example above, but they are produced in such a way that the proof of the equality of each two letters (say, a and b) can be represented as a reasonably small knot-like diagram (with two free ends, labelled a and b), which can be easily inspected by a human for the purposes of checking the proof and finding out interesting facts about the knot diagram. For instance, instead of reading the proofs of $b = g$, $b = d$ and $g = e$ above, one can inspect the diagram in Figure 3; we shall revisit this example in Section 4.

The following sections explain how such labelled tangle diagrams are related to the knot diagram, and why the existence of certain labelled tangle diagrams proves that a knot diagram represents the trivial knot.

Since this research involves considering many types of diagrams, it may be useful to highlight how different diagrams are used for different purposes in this paper.

- There are knot diagrams, such as in Figure 2.
- There are tangle diagrams with two free ends, such as in Figure 3; the existence of certain tangle diagrams with two free ends (such as, for example, in Figures 11–16) proves that the knot diagram is the trivial knot.
- We have proved Theorem 3 which establishes a connection between knot diagrams and tangle diagrams by considering certain more complicated diagrams, which look like tangle diagrams with one special strand which we call a virtual ruler; see Section 3.
- In practice, finding suitable tangle diagrams with two free ends involves considering other tangle diagrams (not only with two free ends), as described in Sections 5 and 6.

2   Groups induced by a knot diagram

This section presents definitions, together with an expanded and corrected exposition of Fact 1 and the subsequent discussion in [11].

By an *arc* we mean a continuous line in a knot diagram from one undercrossing to another undercrossing. For example, consider the knot diagram in Figure 2; it has seven arcs, denoted by $a, b, \ldots, g$.

For a given knot diagram D, the π-orbifold group OD of the knot is a group generated by the arc letters with the following relations. For each arc x of the diagram D, introduce a relation $x^2 = 1$. At every crossing where x and z are the two arcs terminating at the crossing and y is the arc passing over the crossing,

introduce a defining relation $xy = yz$ (or, equivalently, $yx = zy$, or $yxy = z$, or $yzy = x$). Let A denote the generating set of OD (i.e. the set of labels of the arcs of D), and consider the natural homomorphism from the free semigroup A^+ onto OD. It is easy to see that, for each element g of OD, either only words of an odd length are mapped to g or only words of an even length are mapped to g under the homomorphism. Accordingly, let us say that g is an element of odd (even) length in the former (latter) case. A subgroup of OD consisting of the set of all elements of even length is called the fundamental group of the 2-fold branched cyclic cover space of a knot [15, 20]; we shorten this name to the *two-fold group of a knot*, and denote the group by TD.

Another well-known algebraic construction associated with a knot diagram d is its *knot group*, which we denote by GD. This is historically the first and the best known construction (see, for example, Section 6.11 in [6] or Chapter 11 in [10]). We do not need to define GD here, but we note that OD is a factor-group of GD produced from GD by introducing the additional relations $x^2 = 1$ for each arc x.

Trivial knots can be characterised via certain algebraic constructions associated with them.

Theorem 1. *The following are equivalent:*

- *A knot diagram D is a diagram of the trivial knot.*
- *The two-fold group TD is trivial [15, ?]. In other words, for each pair of arc labels x and y, we have $x = y$ in the two-fold group of the knot.*
- *The group GD is infinite cyclic [1]. In other words, for each pair of arc labels x and y, we have $x = y$ in the knot group.*

Since the π-orbifold group OD is 'sandwiched' between the two-fold group of the knot TD and the group of the knot GD, the following conclusion can be made concerning π-orbifold groups.

Corollary 1. *A knot diagram D is a diagram of the trivial knot if and only if its π-orbifold group OD is the two-element cyclic group. In other words, D is a diagram of the trivial knot if and only if for each pair of arc labels x and y, we have $x = y$ in OD.*

The rest of the theoretical discussion in the paper concentrates on discussing how one can prove, for any two given arc labels x and y, that $x = y$ in the π-orbifold group of the knot.

3   Reading Tangles

Tangle diagrams have been used for untangling knot diagrams; see, for example, [8]; in this paper we introduce a completely new way of using tangles to untangle knot diagrams. In this section, we develop the theory of reading tangles. This takes the form of flexible rulers, called virtual rulers, each of which gives rise to a word in the π-orbifold group. Then, the passing of such a ruler through a tangle

diagram corresponds to a proof in the group. We first provide some examples to illustrate the core idea, before moving into the theoretical developments ensuring that the intuitive use of these tangle diagrams and virtual rulers is well-founded.

Figure 4 shows an example of passing a virtual ruler (the line across the tangle, from left to right, with meeting points of the ruler and the diagram accentuated with dots) over a tangle diagram to produce a proof. The tangle diagram has its arcs labeled, and the labels on the ruler concatenate the labels encountered as it passes through the tangle, starting with 1 on the left hand side of the ruler playing the role of the empty word. The tangle diagrams in all six cases in Figure 4 are the same and the rulers differ in a specific way (details to follow). We obtain a proof consisting of equalities of the words from each of the six cases. In this example, reading the words induced by the rulers starting in the top row from left to right, followed by the bottom row gives rise to the proof: $a = cca = cbc = cdb = ccd = d$.
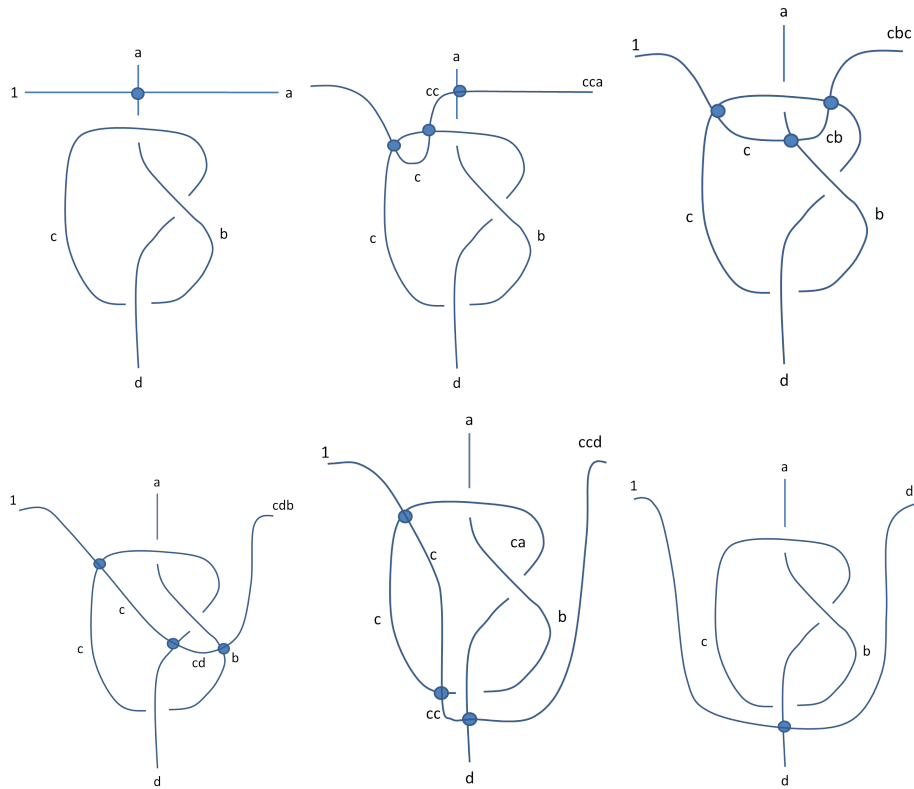
Fig. 4. An example of passing a virtual ruler over a tangle diagram to produce a proof. Reading the words induced by the rulers starting in the top row from left to right, followed by the bottom row gives rise to the proof: $a = cca = cbc = cdb = ccd = d$.

Later on, we will see that each step between diagrams can be viewed precisely as a certain type of move, called a TR_2 or a TR_3 move (these will be defined soon), and we will use this characterization to ensure the equalities we claim in the proof are correct. Alternate choices of application of the TR_2 and TR_3 moves can give rise to different proofs, as demonstrated in Figure 5. The first four diagrams and rulers in the sequences are the same in Figures 4 and 5, but the fifth is different, effectively by applying a different TR_3 move to the fourth diagram in the sequence. From Figure 5, we obtain the slightly different proof: $a = cca = cbc = cdb = dbb = d$.
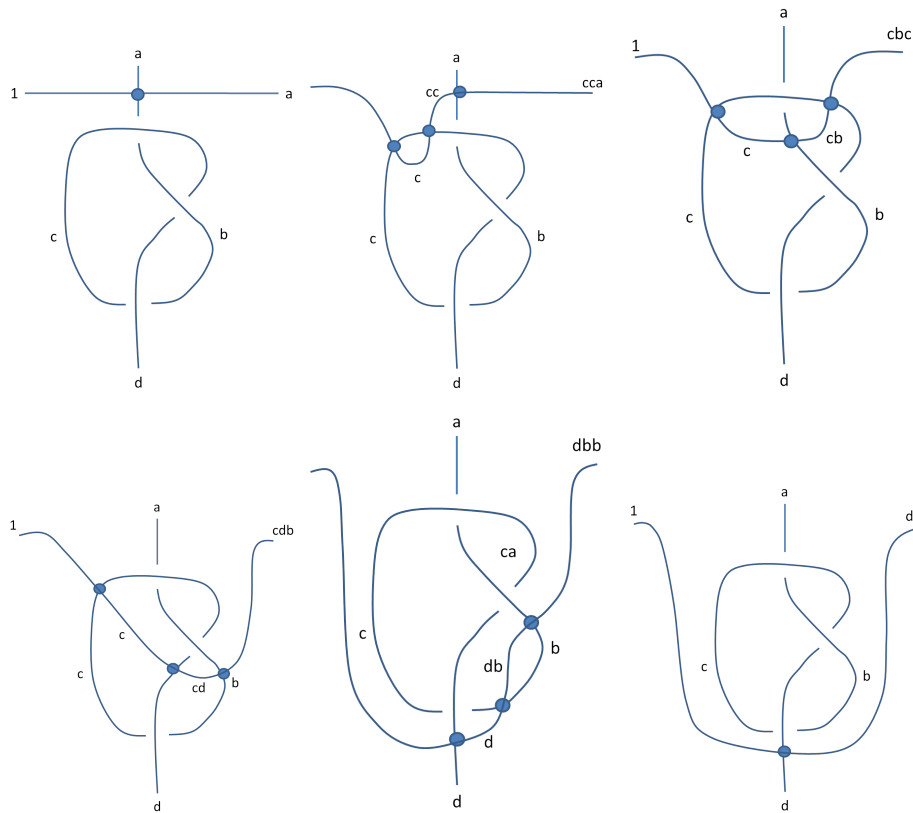

Fig. 5. A different proof obtained by applying a different TR_3 move to the 4th diagram in the sequence in Figure 4. We obtain the proof: $a = cca = cbc = cdb = dbb = d$.

3.1 The theory of reading tangles

A *tangle* diagram T is like a knot diagram, except that its arcs may have free ends (commonly arranged at the top and bottom of a bounding box of the diagram –

this is not essential, as can be seen from some examples in the rest of the paper, but we adopt this convention here to make the exposition more straightforward). The arcs are labeled by elements of the group. For our construction, we require that each labeled crossing matches exactly with one of the labeled crossings of the original knot diagram.

Figure 6 shows an example of a tangle diagram with one free end at the top and one at the bottom. This is the tangle considered in Figures 4 and 5. In this paper, we only need to consider tangles with one free end at top and bottom, but the concept generalizes, as does the theory of reading tangles developed here.
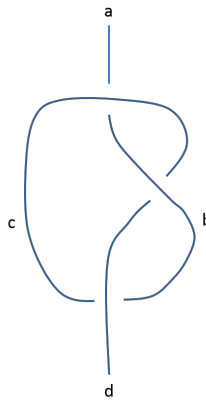
Fig. 6. The tangle diagram which is used in all cases of the Figures 4 and 5 – the additional part of those figures that changed was the virtual ruler.

Definition 1. *A **virtual ruler** v for T is an additional strand with one free end at the left and one at the right of T, which only meets T transversely at points that are not classical crossings.*

This means that the ruler can be viewed as a line drawn through a tangle diagram from the left to the right of the tangle, which crosses the tangle diagram properly (so no tangential meetings or concurrency of line segments) and does not pass through any already existing classical crossing of the tangle diagram. The meeting points between tangle diagram and the ruler are clearly indicated via blobs (which may be called virtual crossings).

Definition 2. *Let T be tangle diagram and v a virtual ruler v for T. Then the arcs of v have labels in the group induced from T by assigning the identity (denoted by 1 here) to the leftmost free arc of v, and concatenating the labels on the arcs of T that v crosses as one traverses the virtual ruler from left to right. The word w obtained as the label on the rightmost free end of v is the **interpretation** of v in T.*

The arcs of v end at the blobs. Here, since we are considering a group we made use of the identity as the leftmost label of the ruler (whilst the empty word ϵ can be used here for the more general setting).

In Figure 7, we present moves of virtual rulers over tangles (the definition follows). These will be precisely the moves of virtual rulers and tangles that encapsulate equivalence (see Theorem 2).
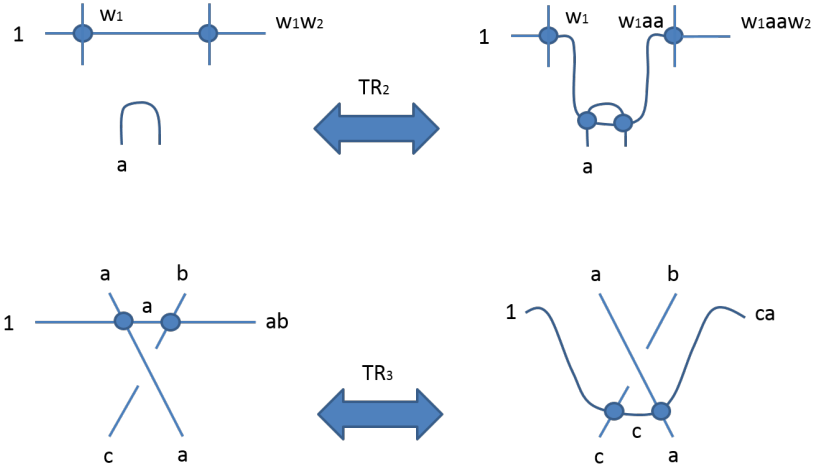

Fig. 7. The TR-moves, indicating permissible moves of a virtual ruler over a tangle.

Definition 3. *Let T be tangle diagram and v a virtual ruler for T. Define two moves of a virtual ruler over T, denoted TR_2 and TR_3, as shown in Figure 7. For TR_2, the w_1 and w_2 are words on the labels of the arcs of the virtual ruler shown; the case in which either (or both) of w_1 and w_2 are empty is also permitted (if the left or right virtual crossing shown, respectively, is not present). For TR_3, the analogous move with a different classical crossing, shown in Figure 8 is also permitted.*

Theorem 2. *Given any two virtual rulers v_1 and v_2 for T, the interpretation of v_1 in T and the interpretation of v_2 in T are equal in OD.*

Proof. In the same way that one can pass a strand of a knot diagram over the rest of the diagram by repeated application of Reidemeister moves R_2 and R_3, one can pass a virtual ruler over T by the repeated application of the TR_2 and TR_3 moves. We see that each of these moves induces an equality in OD by precisely the application of one of the defining relations. For TR_2, it is the application of the relation $a^2 = 1$, with a a generator. For TR_3 it is the relation $ab = ca$, obtained from the labeled crossing in T, which by construction occurred in OD (see the matching crossing which is the second on the left in Figure 10).
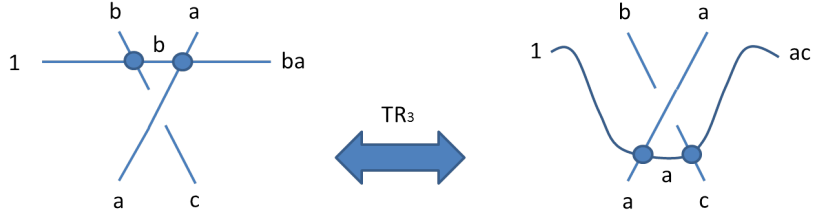
Fig. 8. An analogous T_3 move (we do not distinguish in naming convention).

Lemma 1. *The rules TR_2 and TR_3 are sufficient to pass a virtual ruler over a tangle.*

Proof. Consider the stepwise process of moving a virtual ruler over a fixed tangle by moving *segments* down the tangle from the top to the bottom. By *segment* here we refer to a connected part of the virtual ruler, not necessarily starting or ending at virtual crossings, which contains at most two virtual crossings. Any such segment considered contains 0, 1 or 2 virtual crossings. In the case of 0 virtual crossings, only TR_2 can be applied (see top row of Figure 9). In the case of 2 virtual crossings, either we are in the case where TR_3 can be directly applied (if there is a Δ with the two virtual crossings and a classical crossing at the corners and there are no other strands meeting the Δ, as in the middle row of Figure 9), or not. If not, then consider a segment containing only one of the strands meeting the virtual ruler at one of these two virtual crossings, reducing to the case of 1 virtual crossing. In this case, apply a nearby TR_2 move, which enables the subsequent application of a TR_3-move (see bottom row of Figure 9). Any TR_3-move applied reduces the number of classical crossings remaining to pass over. The process terminates.

Recall that a *proof* in OD is a sequence of equalities of elements of OD such that each consecutive element differs by the application of one of the relators in OD.

Proposition 1. *Let T be a tangle diagram, and let $v_1, \ldots, v_k$ be a sequence of virtual rulers for T, with interpretations $i(v_1), \ldots, i(v_k)$. Then:*

1. *The sequence of equalities $i(v_1) = \ldots = i(v_k)$ holds in OD.*

2. *If, in addition, for each $j \in \{1, \ldots, k-1\}$ we have that v_j differs from v_{j+1} by a TR move (either TR_2 or TR_3), then the sequence of equalities $i(v_1) = \ldots = i(v_k)$ is a proof in OD.*

Proof. The first part follows from Theorem 2, whilst the second part follows from the observation in the proof of Theorem 2 about the matching of the TR moves with the relators.
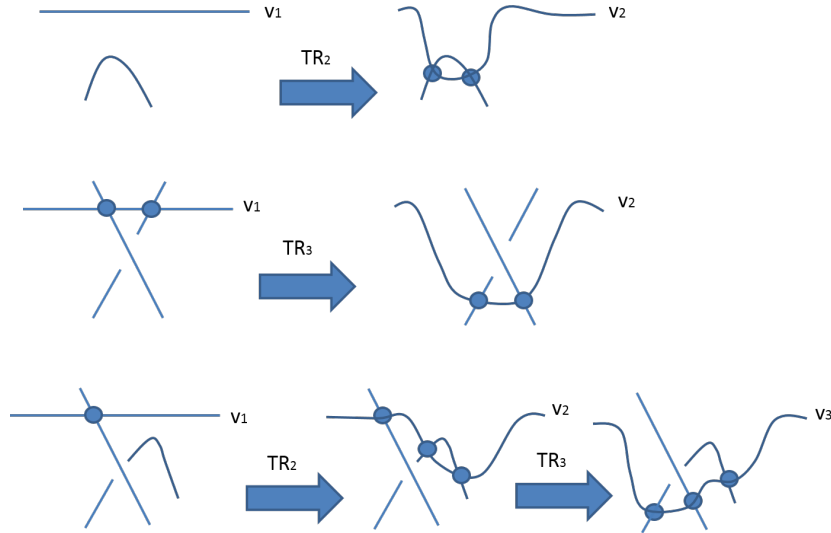
Fig. 9. The process of moving a virtual ruler over a tangle.

Corollary 2. *There exist different proofs obtained from one fixed T corresponding to different sequences of virtual rulers over T.*

Proof. See Figures 4 and 5.
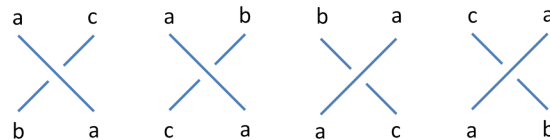
Fig. 10. Relations, from left to right: $ac = ba, ab = ca, ba = ac, ca = ab$.

4   Untangling: the main result and an example

The theory we developed gives us a practical method for proving that a knot diagram is a diagram of the trivial knot by drawing certain tangle diagrams.

Theorem 3.   *A knot diagram D (with unique labels) represents the trivial knot if and only if for each pair of its labels a, b there is a labelled tangle diagram T*

which has exactly two free-end arcs labelled a and b, with the property that each crossing in T is labelled in the same way as some crossing in D.

Proof. The 'if' direction follows directly from Corollary 1 and Theorem 2. The 'only if' direction follows from the fact that every derivation of an equality of two letters $x = y$ in OD naturally induces a tangle diagram with two free ends x and y, with relations of OD being transformed into crossings as on Figure 10.

We present a small example, produced manually. In the next section we report on our progress with building tangle diagrams using the computer.

*Example 1.* The knot diagram in Figure 2 represents the trivial knot.

The proof splits into several lemmas, each demonstrating that two arc labels (that is, two generators of OD) are equal. The lemmas and their proofs are presented on Figures 11–16; note that these diagrams are not illustrations of proofs, but actual proofs: that is, the existence of a diagram presented on Figure 11 is, according to Theorem 3, a proof that $b = g$, and so on. For brevity of presentation, in some of the lemmas, a rectangle marked i is used as shorthand, meaning that the diagram from Lemma i should be substituted for this rectangle. This convention enables us to make diagrams more compact. As an example of the use of rectangles, compare the diagram in Lemma on Figure 13 with the equivalent diagram in Figure 3, which presents the same step in the proof. Whilst the equalities are immediate from the earlier results, the interested reader can also directly compare the equalities read off from each tangle diagram as one traverses from one free end to the other (e.g. from b to g on Figure 13) and compare with the algebraic proof presented earlier (e.g. "$bc = ca = gc$, hence, $b = g$").
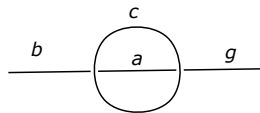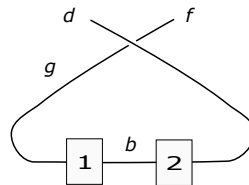

Fig. 11. $b = g$
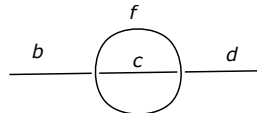
Fig. 12. $b = d$
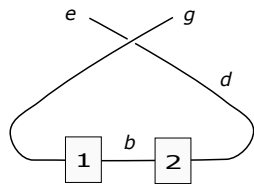
Fig. 13. $e = g$

Fig. 14. $d = f$
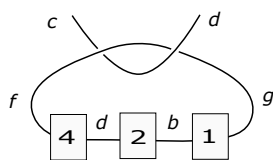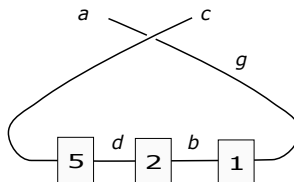
Fig. 15. $c = d$

Fig. 16. $a = c$

5 Manipulating tangle diagrams with the computer

One can observe that the tangle diagrams in proofs (for example, like those in Figure 3) may be assembled from copies of individual crossings of the original knot diagram by applying steps of the following two types:

— Given a labelled tangle diagram which has, among its end arcs, two adjacent end arcs labelled with the same letter, connect these two arcs.
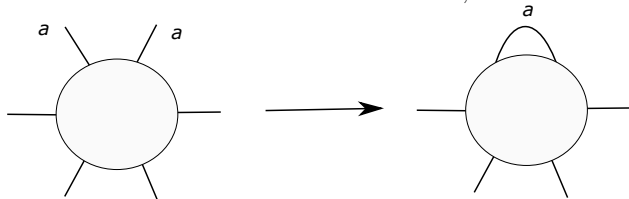
— Given two labelled tangle diagrams T and U such that both T and U have an end arc labelled with the same letter, connect these two arcs.
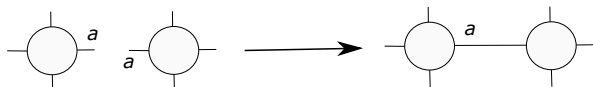
To do this using the computer, instead of tangle diagrams themselves, we consider words which one can read on end arcs around a tangle diagram (clockwise or anticlockwise, starting from any point outside the diagram). To start with, we list all words which one can read around individual crossings of the knot diagram such as, for example, $acbc$ around the top crossing of the diagram on Figure 2 (for the top crossing walking in a small circle around the crossing meet four arcs and the labels are recorded in the order the arcs are met). Whenever we have a word $a_1 a_2 \ldots a_{n-1} a_n$, we also produce the word $a_2 \ldots a_{n-1} a_n a_1$ (which is a cyclic shift of the original word, corresponding graphically to starting reading from a different point outside the diagram) and the word $a_n a_{n-1} \ldots a_2 a_1$ (which is the original word inverted, corresponding graphically to reading around the tangle in the opposite direction)[4]. In addition to these two 'trivial' ways of producing new words, we have two more, corresponding to the graphical moves above. Whenever we have a word $a_1 a_2 a_3 \ldots a_n$ and $a_1 = a_2$, we produce the

---

[4] In the implementation presented in Section 6 we did not use word reversion because, although reversing a word may make a proof shorter, it is not easy to implement it in the prover software we used.

word $a_3 \dots a_n$ (which corresponds to connecting two adjacent arcs with identical labels). Whenever we have two words $a_1 a_2 \dots a_n$ and $b_1 b_2 \dots b_n$ and $a_1 = b_1$, we produce the word $a_2 \dots a_n b_2 \dots b_n$ (which corresponds to connecting two tangle diagrams)[5].

The aim of this process is to produce all (or, to be more precise, 'sufficiently many') two-letter words ab. In our computer experiments (as presented in Section 6), we enumerated all arc labels of the knot diagram in some order and proved each equality $a_i = a_{i+1}$ by proving that we can produce the word $a_i a_{i+1}$.

6 Automated Proofs

We experimented with the procedure described in Section 5 by using its reduction to automated theorem proving. To a given knot diagram D we can associate a first-order theory T_D in a vocabulary which consists of unary predicate symbol T, binary functional symbol $*$ and constants e and $a_1, \dots, a_k$ for all of the labels of D. The axioms of T_D include:

I. Axioms of monoid for $(*, e)$:
- $(x * y) * z = x * (y * z)$ (associativity of multiplication)
- $x * e = e * x = x$ (e is a unit of the monoid)

II. - $a_i^2 = e$ for all labels a_i

III. Initial state axioms:
- $T(a_i * a_j * a_k * a_j)$ for all crossings in D, where the over-crossing arc is labelled by a_j and the under-crossing arcs are labelled by a_i and a_k.

IV. Transition axioms:
- $T(a_i * x) \to T(x * a_i)$ for all arc labels a_i
- $T(x) \;\&\; T(y) \to T(x * y)$.

The ground terms are built from constants by the monoid operation, and are meant to represent words read on the ends of arcs around the tangle diagrams. The intended meaning of $T(w)$ for a word w is a corresponding diagram (i.e with w read on its end arcs) which can be built following the rules in Section 5. The initial state axioms declare that the original crossings present us with the initial building blocks of the tangle diagrams to be able to start construction. The transition axioms describe permissible operations for building new tangles. The following result holds.

Proposition 2. *A knot diagram D with arcs labelled by $a_1, \dots, a_k$ is a diagram of trivial knot (unknot) if and only if $T_D \vdash \bigwedge_{1 \leq i \leq k-1} T(a_i * a_{i+1})$, where $\vdash$ denotes first-order logic derivability.*

---

[5] Another approach that we tested (as presented in Section 6) is simply to place two tangle diagrams next to each other without connecting them by an arc; this corresponds, in terms of labelling words, to concatenating the words.

Proof. Due to Theorem 3 it is sufficient to show that $T_D \vdash T(a_i * a_{i+1})$ iff a labelled tangle diagram can be built for D with two free end arcs labelled by a_i and a_{i+1} using the procedure from Section 5.

$\Leftarrow$: Assume that a tangle diagram with free ends labelled by a word w is built by the procedure. Then straightforward induction on the length of construction shows that $T_D \vdash T(t_w)$, where t_w is a term encoding of w, that is a term built from $a_1, \ldots, a_k$ using $*$. Indeed, the statement holds for initial tangle diagrams formed by original crossings due to axioms **III**, which are included in T_D. For the inductive step, we assume that the statement holds for two tangle diagrams labelled by words w_1 and w_2, so we have that $T_D \vdash T(t_{w_1})$ and $T_D \vdash T(t_{w_2})$. Then, if w' labels a tangle diagram obtained by connecting two adjacent arcs in the tangle labelled by w_1 we have that $T_D \vdash T(t_{w'})$, using the induction hypothesis for w_1 and the axioms **II** and **I**. If a tangle diagram labelled by w'' is obtained by connecting tangle diagrams labelled by w_1 and w_2 then $T_D \vdash T(t_{w''})$ using the inductive hypothesis for w_1 and w_2 and the transition axioms **IV**.

$\Rightarrow$: By induction on the number of applications of *Modus Ponens* rule using a transition axiom $T(x) \& T(y) \rightarrow T(x * y)$ we show that if $T_D \vdash T(t_w)$ then a tangle diagram can be constructed by the procedure from Section 5, which is labelled by w' such that $\mathbf{I}, \mathbf{II} \vdash t_w = t_{w'}$. Indeed, for the base of induction we notice that the only formulae of the form $T(\ldots)$ derivable form T_D without applying *Modus Ponens* to $T(x) \& T(y) \rightarrow T(x * y)$ are formulae $T(t_{w'})$ with $t_{w'}$ such that $\mathbf{I}, \mathbf{II} \vdash t_w = t_{w'}$ for some t_w from an initial state axiom $T(t_w)$. Then the crossing corresponding to this axiom provides with the required tangle.

For the step of induction consider a derivation $T(t_w * t_{w'})$ from already derived $T(t_w)$ and $T(t_{w'})$ and the transition axiom $T(x) \& T(y) \rightarrow T(x * y)$. By the induction assumption we have required tangle diagrams constructed for $T(t_w)$ and $T(t_{w'})$, that is diagrams labelled by w and w', respectively. Then the tangle diagram for $T(t_w * t_{w'})$ is constructed by placing latter diagrams next to each other, and it is labelled by ww'. Further derivations of formulae of the form $T(..)$ from $T(t_w * t_{w'})$ using axioms $\mathbf{I}, \mathbf{II}$ and first transition axiom are possible. In all such cases the required tangle diagram will be either the same as for $T(t_w * t_{w'})$, or reduced by connecting two adjacent arcs with equal labels.

Proposition 2 suggests a procedure for establishing unknottedness by using automated provers for first-order logic. Given a knot diagram D, specify a theory T_D and apply an automated theorem prover to $T_D \vdash \bigwedge_{1 \le i \le k-1} T(a_i * a_{i+1})$.

In Table 1 we report on experiments[6] using the automated prover Prover9 [14] on a few well-known unknot diagrams.

The authors have used the same software to implement other algebraic techniques (similar to the one described in Corollary 1) for proving unknottedness of knot diagrams [4], [11]. For the diagrams listed in Table 1, other techniques prove unknottedness faster (as expected). However (also as expected), the proofs of unknottedness produced by the methods proposed in this paper are more transparent and more amenable to the interpretation as untangling sequences.

---

[6] System used in experiments: Intel(R) Core(TM) i7-4790 CPU 3.60Ghz, RAM 32 GB, Windows 7 Enterprise

Table 1. Time taken to prove unknottedness of some known diagrams

Name of unknot	Reference	# of crossings	Time, s
"Trivial" Trefoil	[8]	3	0.01
No Name	Fig.2	7	0.09
Culprit	[8]	10	5.16
Goerlitz	[7]	11	6.38
Thistlethwaite	[18]	15	321.0
Ochiai, I	[17]	16	1286.1

7 Conclusion & Future work

In the paper, we proposed a novel use of labelled tangle diagrams as a means of representing certain algebraic proofs. This provides the opportunity for providing a visual readable proof instead of an algebraic textual proof, such that the proof notation is of a similar type to the original type of diagrams considered. We developed theory to demonstrate the equivalence of the use of these tangle diagrams to express correct algebraic equalities, in effect showing that proofs in the algebra can be represented visually via these labelled tangles. This was performed making use of some new, reusable machinery (virtual rulers, TR-moves) that will have independent uses. Whilst such proofs for small scale diagrams can be manually developed (as per the example in Section 4), there is a definite need for computer assistance on the larger scale. To this end, we present progress in automating the production of such proofs. We provide an indication of the time taken for the automate search for such proofs, and we observe a likely trade-off in the form of a slow-down of unknot detection in order to be able to develop the new visual proofs rather than algebraic proofs only.

This work opens us several avenues for future research. Firstly, extracting readable proofs from the output of a general-purpose prover software is very difficult. To help with this task, after the prover software has produced proofs, another script is needed to convert the proofs into a readable form. Alternatively, we can write our own specialist prover code. Secondly, producing nice-looking tangle diagrams from proofs manually is time consuming. For larger diagrams, the actual diagrams must be produced by the computer. After such tool support is present, an exploration of the utility of the representation and the tool by different possible user groups (e.g. undergraduate students versus research mathematicians) will also become feasible.

References

1. Gerhard Burde, Michael Heusener, and Heiner Zieschang. *Knots*. De Gruyter, 2013.
2. Alexander Coward and Marc Lackenby. An upper bound on reidemeister moves. *American Journal of Mathematics*, 136(4):1023–1066, 2014.
3. IVAN A Dynnikov. Three-page approach to knot theory. Universal semigroup. *Functional Analysis and Its Applications*, 34(1):24–32, 2000.

4. Andrew Fish and Alexei Lisitsa. Detecting unknots via equational reasoning, i: Exploration. In *International Conference on Intelligent Computer Mathematics*, pages 76–91. Springer, 2014.

5. Andrew Fish, Alexei Lisitsa, and David Stanovský. A combinatorial approach to knot recognition. In Ross Horne, editor, *Embracing Global Computing in Emerging Economies: First Workshop, EGC 2015, Almaty, Kazakhstan, February 26-28, 2015. Proceedings*, pages 64–78. Springer International Publishing, 2015.

6. ND Gilbert and Timothy Porter. *Knots and surfaces*. Oxford University Press, 1994.

7. Louis H. Kauffman and Allison Henrich. Unknotting unknots. https://arxiv.org/abs/1006.4176.

8. Louis H. Kauffman and Sofia Lambropoulou. Hard unknots and collapsing tangles. https://arxiv.org/abs/math/0601525v5.

9. Akio Kawauchi. *A survey of knot theory*. Birkhäuser, 1996.

10. WB Raymond Lickorish. *An introduction to knot theory*, volume 175. Springer Science & Business Media, 2012.

11. Alexei Lisitsa and Alexei Vernitski. Automated reasoning for knot semigroups and π-orbifold groups of knots. In *International Conference on Mathematical Aspects of Computer and Information Sciences*, pages 3–18. Springer, 2017.

12. Charles Livingston. *Knot theory*, volume 24. Cambridge University Press, 1993.

13. Vassily Manturov. *Knot theory*. CRC press, 2004.

14. W. McCune. Prover9 and mace4. `http://www.cs.unm.edu/~mccune/prover9/`, 2005–2010.

15. John W Morgan and Hyman Bass, editors. *The Smith Conjecture*. Elsevier, 1984.

16. Kunio Murasugi. *Knot theory and its applications*. Springer Science & Business Media, 2007.

17. Mitsuyuki Ochiai. Non-trivial projections of the trivial knot. http://repository.kulib.kyoto-u.ac.jp/dspace/handle/2433/99940.

18. Unknot. https://en.wikipedia.org/wiki/Unknot.

19. Alexei Vernitski. Describing semigroups with defining relations of the form xy= yz and yx= zy and connections with knot theory. *Semigroup Forum*, 95(1):66–82, 2017.

20. Steven K. Winker. Quandles knot invariants and the n-fold branched cover. Technical report, 1984.