The Word Problem for Braided Monoidal Categories is Unknot-Hard

Antonin Delpeuch
Department of Computer Science, University of Oxford
antonin.delpeuch@cs.ox.ac.uk

Jamie Vicary
Computer Laboratory, University of Cambridge
jamie.vicary@cl.cam.ac.uk

We show that the word problem for braided monoidal categories is at least as hard as the unknotting problem. As a corollary, so is the word problem for Gray categories. We conjecture that the word problem for Gray categories is decidable.

Introduction

The *word problem* for an algebraic structure is the decision problem which consists in determining whether two expressions denote the same elements of such a structure. Depending on the equational theory of the structure, this problem can be very simple or extremely difficult, and studying it from the lens of complexity or computability theory has proved insightful in many cases.

This work is the third episode of a series of articles studying the word problem for various sorts of categories, after monoidal categories [6] and double categories [5]. We turn here to braided monoidal categories. Unlike the previous episodes, we do not propose an algorithm deciding equality, but instead show that this word problem seems to be a difficult one. More precisely we show that it is at least as hard as the unknotting problem.

The unknotting problem consists in determining whether a knot can be untied and was first formulated by Dehn in 1910 [3]. The decidability of this problem remained open until Haken gave the first algorithm for it in 1961 [9]. As of today, no polynomial time algorithm is known for it.

One reason why we are interested in braided monoidal categories is that they are a particularly natural sort of categories, in the sense that they arise as doubly degenerate Gray categories [8]. Studying the word problem for them is therefore a first step towards understanding the word problem for weak higher categories, for which little is known to date.

This article starts off with a section giving some background on the various flavours of monoidal categories we will use, as well as a quick introduction to the unknotting problem. Then, we give a first reduction between the unknotting problem to the braided pivotal word problem, as a way of introducing tools which will be needed for the last section, where our main result is proved.

Acknowledgements

The authors wish to thank the participants of the 2019 Postgraduate Conference in Category Theory and its Applications in Leicester and Makoto Ozawa for their feedback and help on this project. The first author is supported by an EPSRC scholarship.

1 Background

We assume familiarity with monoidal categories and their string diagrams [19].

K. Kishida (Ed.): Fourth International Conference
on Applied Category Theory (ACT 2021).
EPTCS 372, 2022, pp. 72–87, doi:10.4204/EPTCS.372.6

(a) String diagram for $\sigma_{A,B}$ (b) String diagram for $\sigma_{A,B}^{-1}$

Figure 1: Representation of braid morphisms as string diagrams

Figure 2: The hexagon identities represented as string diagrams

1.1 Braided monoidal categories

Braided monoidal categories were introduced by [11, 13]. In this work, we study the word problem for this algebraic structure. This is the decision problem where given two expressions of morphisms in a free braided monoidal category, we need to determine whether or not they represent the same morphism.

In what follows, all monoidal categories are strict. There exists weak versions of the following definitions, and coherence theorems show their equivalence with the strict definitions that we use here. See for instance Theorem 4 in [11] for the case of braided monoidal categories.

Definition 1. *A* ***braided monoidal category*** $\mathscr{C}$ *is a monoidal category* $(\mathscr{C}, \otimes, I)$ *equipped with a natural isomorphism* $\sigma_{A,B} : A \otimes B \to B \otimes A$*, satisfying the hexagon identities:*

$$\sigma_{A,B\otimes C} = (1_B \otimes \sigma_{A,C}) \circ (\sigma_{A,B} \otimes 1_C)$$
$$\sigma_{A\otimes B,C} = (\sigma_{A,C} \otimes 1_B) \circ (1_A \otimes \sigma_{B,C})$$

We use string diagrams for monoidal categories to represent morphisms in braided monoidal categories. Figure 1 shows the representation of the braid morphism and its inverse. Figure 2 shows the representation of the hexagon identities with this convention.

The soundness and completeness theorem of string diagrams for monoidal categories can be extended to the case of braided monoidal categories [12].[1] This requires adapting the notion of string diagram, which is now three-dimensional, and the corresponding class of isotopies. We state the soundness and

[1] Soundness and completeness theorems for graphical languages are sometimes called *coherence theorems* but we avoid this terminology because of the confusion it creates with other "coherence" theorems.

(a) Reidemeister 2 move (σ is an iso) (b) Pull-through move (naturality of σ)

Figure 3: Equalities satisfied by braid morphisms

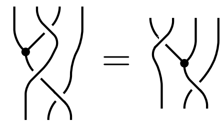
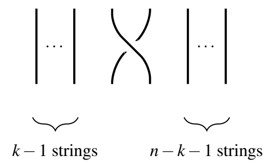
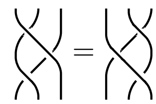


Figure 4: Two regularly isotopic diagrams in a braided monoidal category



(a) Representation of σ_k



(b) Representation of Equation 1

Figure 5: Graphical representation for the braid group

completeness theorem as formulated by [19]. To tell it apart from other notions of isotopy, we use the name *regular isotopy* for the class of isotopies used in this theorem.

Theorem 2. *A well-formed equation between morphisms in the language of braided monoidal categories follows from the axioms of braided monoidal categories if and only if it holds in the graphical language up to regular isotopy in 3 dimensions.*

The combinatorics of string braidings have been studied extensively, but more often from the perspective of group theory than category theory. The braid group was introduced by [1].

Definition 3. *The braid group on n strands B_n, is the free group generated by generators $\sigma_1, \dots, \sigma_{n-1}$ with equations*

$$\sigma_k \sigma_{k+1} \sigma_k = \sigma_{k+1} \sigma_k \sigma_{k+1} \qquad \textit{for } 1 \leq k < n-1 \tag{1}$$

$$\sigma_i \sigma_j = \sigma_j \sigma_i \qquad \textit{for } j - i > 1 \tag{2}$$

for $1 \leq k < n-1$.

With this formalism, an element of the group represents a braid on n strings. The element σ_k represents a positive braiding of the adjacent strings k and $k+1$, and no change on other strings. Its inverse σ_k^{-1} is the negative braiding on the same strings. Figure 5 shows how generators and equations of the braid group can be represented graphically. Equation 1 is called the Reidemeister type 3 move (or Yang-Baxter equation). One can see that it is a particular case of the pull-through move of Figure 3b, where the morphism being pulled through is a braid itself. Equation 2 corresponds to the exchange move in monoidal categories.

We can make the connection between this group-theoretic presentation and braided monoidal categories more precise.

Definition 4. *A **monoidal signature** (G, M) is a set of generating objects G and a set of generating morphisms M. Each generating morphism is associated with two lists of generating objects* $\mathrm{dom}(M)$ *and* $\mathrm{cod}(M)$*, each being elements of G^*.*

Given a monoidal signature, one can generate the free braided monoidal category on it. By Theorem 2, this is the category of braided string diagrams whose vertices and edges are labelled by generating objects and morphisms respectively. Note that we are not imposing any additional equation between the generators: the only equations which hold are those implied by the braided monoidal structure itself.

Proposition 5 [13]. *The free braided monoidal category $\mathscr{B}$ generated by the signature $(\{A\},\emptyset)$ is the braid category, i.e. $\mathscr{B}(A^n,A^n) = B_n$, the group of braids on n strands.*

This can easily be seen in string diagrams: a morphism in $\mathscr{B}$ can only be made of identities, positive and negative braids. As a string diagram in a monoidal category, it can be drawn in general position, where all braids appear at a different height. This can therefore be decomposed as a sequential composition of slices containing exactly one positive or negative braid. The number of wires between each slice remains constant given that braids and identities always have as many outputs as inputs; let us call this number n. Each of these slices corresponds to a generator or generator inverse in B_n. The equations holding in $\mathscr{B}(A^n,A^n)$ and B_n are the same, hence the equality.

Therefore, braided monoidal categories generalize the braid group by allowing for other morphisms than braids and identities. The word problem for the braid group is well understood:

Theorem 6. *The word problem for the braid group B_n can be solved in polynomial time: given two strings of generators and generator inverses, one can determine if they represent the same braid in quadratic time in the length of the strings (for a fixed n).*

See [4] for a review of the various techniques which can be used to achieve this complexity. It seems hard to generalize any of these to the case of braided monoidal categories.

Intuitively, the word problem for braided monoidal categories is harder than the one for the braid group because of the existence of other morphisms which can block the interaction between braids. Because of these additional morphisms, string diagrams in braided monoidal categories can look *knotted* and the equivalence problem for them intuitively becomes harder. In this paper we make this intuition more precise, by showing that the equivalence problem for braided monoidal categories is at least as hard as the unknotting problem.

1.2 Braided pivotal categories

Definition 7. *In a monoidal category $\mathscr{C}$, an object $A \in \mathscr{C}$ has a* ***left adjoint*** *$B \in \mathscr{C}$ (or equivalently, A is the* ***right adjoint*** *of B) when there are morphisms* 

and 

such that the ***yanking equations*** *(or* ***zig-zag equations****) are satisfied:*

Definition 8. *A monoidal category $\mathscr{C}$ is* ***left autonomous*** *if every object $A \in \mathscr{C}$ has a left adjoint ${}^*A$. A monoidal category $\mathscr{C}$ is* ***right autonomous*** *if every object $A \in \mathscr{C}$ has a right adjoint A^*. A category that is both left and right autonomous is simply called* ***autonomous****.*

Lemma 9. *Any braided monoidal category that is left autonomous is also right autonomous (and therefore autonomous).*

Proof. See Lemma 4.17 in [19]. □

Definition 10. *A* ***strict pivotal category*** *is a monoidal category where every object A has identical left and right adjoints.*

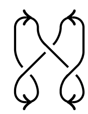
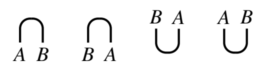


Figure 6: An oriented knot as a morphism in $\mathbb{ROTang}$

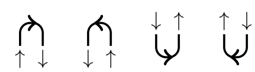
In a braided pivotal category, for each object A there is an object B with the following morphisms:



such that all four yanking equations are satisfied.

Definition 11 [7]. $\mathbb{ROTang}$ *is the free braided pivotal category generated by an object represented by the symbol "↑". We denote by "↓" its adjoint.*

As the notations suggest, the wires of string diagrams in $\mathbb{ROTang}$ can be associated with an upwards or downwards orientation. We adopt the following representation for the morphisms arising from the pivotal structure:

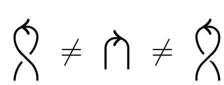


With this convention, we can represent any oriented knot, i.e. any embedding of an oriented loop in $\mathbb{R}^3$, as a morphism in $\mathbb{ROTang}$, as in Figure 6. In fact, this representation is more than a convention, as the following theorem shows:

Definition 12 (Definition 2.4 in [7]). *A* tangle *is a rectangular portion of a knot diagram. We say that two tangles are equal if there is a regular isotopy carrying one to the other in such a way that corresponding edges of the diagram are preserved setwise.*

Theorem 13 (Theorem 3.5 in [7]). $\mathbb{ROTang}$ *is the category of oriented tangles up to regular isotopy.*

This means that two morphisms $f, g \in \mathbb{ROTang}$ are equal if and only if their string diagrams, considered as oriented tangles in three-dimensional space as defined above, are regularly isotopic. Regular isotopy is a more restrictive sort of isotopy than the notion generally used in knot theory, as the following morphisms are distinct in $\mathbb{ROTang}$:



The move that equates them is called the Reidemeister type I move, which is therefore not admissible for the string diagrams of $\mathbb{ROTang}$.

Definition 14. *A morphism $f \in \mathbb{ROTang}$ is an* ***(oriented) knot*** *if its string diagram has a single connected component.*

1.3 The unknotting problem

In this section we give a brief overview of the unknotting problem and some complexity results about it.

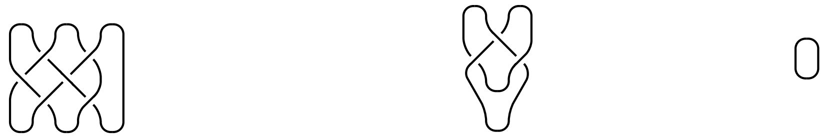
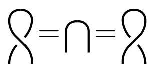
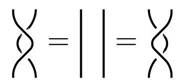


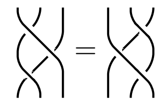
Figure 7: Some knot diagrams



(a) Reidemeister type I



(b) Reidemeister type II



(c) Reidemeister type III

Figure 8: Reidemeister moves

Definition 15. *A* ***topological knot*** *is the embedding of a loop in* $\mathbb{R}^3$. *Two knots* K_1, K_2 *are in* ***general isotopy*** *if there is an orientation-preserving homeomorphism* h *of* $\mathbb{R}^3$ *such that* $h(K_1) = h(K_2)$. *A* ***knot diagram*** *is the projection of a knot on a plane, such that no two crossings happen at the same place. Additionally the diagram records the relative position of the strands at each crossing.*

All knots considered here will be required to be tame, i.e. isotopic to a polygonal knot. This gets rid of some pathological cases.

Some example knot diagrams are given in Figure 7. The Reidemeister moves are local transformations of knot diagrams which are divided in three categories, as shown in Figure 8. Note that in addition to these moves, all planar isotopies are implicitly allowed, without restricting the direction of strands in any way (unlike the recumbent isotopies of monoidal string diagrams).

Theorem 16 (Reidemeister). *Two knot diagrams represent topological knots in general isotopy if and only if they are related by a sequence of Reidemeister moves.*

Knot diagrams can be encoded in various ways, for instance as four-valent planar maps where vertices are crossings and edges are parts of strands. This makes it possible to formulate decision problems about topological knots and study their complexity.

Definition 17. *The unknotting problem* UNKNOT *is the decision problem to determine if a topological knot can be related by a general isotopy to the unknot. In other words, it consists in determining whether there exists a series of Reidemeister moves which eliminates all crossings in a given knot diagram.*

This problem was first formulated by [3] and its decidability remained open until [9] found an algorithm for it. The problem has since attracted a lot of attention and we give a summary of the latest results about it.

Theorem 18 [14]. *There exists a polynomial* $P(c)$ *such that for every knot diagram* K *of the unknot with* c *crossings, there is a sequence of Reidemeister moves unknotting it whose length is bounded by* $P(c)$.

Corollary 19. UNKNOT *lies in NP.*

Theorem 20 [15]. UNKNOT *lies in co-NP.*

Recently, Lackenby announced a quasi-polynomial time solution to UNKNOT, but the corresponding article has not been made public to date. No polynomial time algorithm for this problem is known so far.

2 Reducing the unknotting problem to the braided pivotal word problem

Despite the discrepancy between the general isotopy used in the unknotting problem and the regular isotopy used in $\mathbb{ROTang}$, we will show that the unknotting problem can be reduced to the word problem in $\mathbb{ROTang}$. This will show that the word problem for $\mathbb{ROTang}$ is at least as hard as the unknotting problem. This section is dedicated to this result.

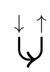
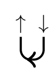
2.1 Writhe and turning number

The main differences between the unknotting problem and the word problem for $\mathbb{ROTang}$ is that in the latter, knots are oriented and the Reidemeister type I move is not allowed. Because of this, we will see in this section that we can associate a quantity called *writhe* to diagrams in $\mathbb{ROTang}$, which is preserved by all the axioms of this category.

Definition 21. *The **writhe** (or* framing number*) $W(f)$ of a diagram $f \in \mathbb{ROTang}$ is the sum of the valuations $W(b)$ for each braiding b which appears in f:*

$W($  $) = +1$ $W($  $) = -1$

Definition 22. *The **turning number** (or* winding number*) $T(f)$ of a morphism $f \in \mathbb{ROTang}$ is the sum of the local turning numbers which appear in f:*

$T($  $) = +1$ $T($  $) = -1$ $T($  $) = -1$ $T($  $) = +1$

The turning number is well defined because the axioms of $\mathbb{ROTang}$ respect the turning number, making it independent of the particular diagram considered.

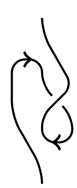
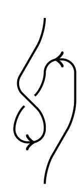
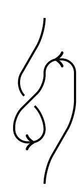
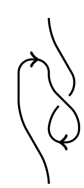
Theorem 23 [20]. *Let $f, g \in \mathbb{ROTang}$ be two knots. Then f and g are in general isotopy as knots (allowing Reidemeister type I moves) if and only if $W(f) = W(g)$, $T(f) = T(g)$ and there is a regular isotopy between f and g (disallowing Reidemeister type I moves).*

This means that to reduce the unknot problem to the word problem for $\mathbb{ROTang}$, we simply need to be able to tweak diagrams to adjust their writhe and turning number without changing their general isotopy class. This is what the following section establishes.

2.2 Unknotting in braided pivotal categories

Lemma 24. *Given a writhe w and a turning number t such that $2w+t$ is a multiple of 4, we can construct a morphism $f \in \mathbb{ROTang}$ with domain and codomain $\uparrow$, such that $W(f) = w$ and $T(f) = t$, and f is in general isotopy to the identity using Reidemeister type I moves.*

Proof. We first define the following morphisms in $\mathbb{ROTang}(\uparrow, \uparrow)$:

$a =$  $b =$  $c =$  $d =$ 

They have the following invariants:

$$W(a) = +1 \qquad W(b) = -1 \qquad W(c) = +1 \qquad W(d) = -1$$
$$T(a) = +2 \qquad T(b) = -2 \qquad T(c) = -2 \qquad T(d) = +2$$

Let $w, t \in \mathbb{Z}$ such that $2w+t$ is a multiple of 4. We construct the required morphism $f \in \mathbb{ROTang}(\uparrow,\uparrow)$ by composition of a, b, c and d using the fact that $W(g \circ h) = W(g) + W(h)$ and $T(g \circ h) = T(g) + T(h)$ for all $g, h \in \mathbb{ROTang}(\uparrow,\uparrow)$. Let $p = \frac{2w+t}{4}$. If p is positive, we start by p copies of a, otherwise $-p$ copies of b. Then, let $q = \frac{2w-t}{4}$. If q is positive, we continue with q copies of c, otherwise $-q$ copies of d. One can check that the composite has the required writhe and turning number. ☐

Corollary 25. *The general isotopy problem for knots can be reduced to the word problem for* $\mathbb{ROTang}$.

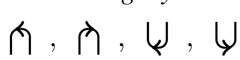
Proof. Given two knots k, l represented as crossing diagrams in the plane, pick an orientation for them and turn them into morphisms $f, g \in \mathbb{ROTang}$. We can compute the writhe and turning number of f and g in polynomial time.

As noted by [20], for any oriented knot f, $\frac{2W(f)+T(f)}{2}$ is odd. In other words there are $p, q \in \mathbb{Z}$ such that $2W(f) + T(f) = 4p+2$ and $2W(g) + T(g) = 4q+2$. Therefore $2(W(f) - W(g)) + (T(f) - T(g)) = 4(p-q)$. By Lemma 24, we can therefore construct a morphism $h \in \mathbb{ROTang}(\uparrow,\uparrow)$ such that $W(h) = W(f) - W(g)$ and $T(h) = T(f) - T(g)$, and such that h can be related by a general isotopy to a straight wire (so, allowing Reidemeister type I moves).

Therefore we can insert h on any strand of g, obtaining a morphism g' which represents the same knot as g, such that $W(g) = W(f)$ and $T(g) = T(f)$. By Theorem 23, f and g' are in general isotopy as knots if and only if they are equal as morphisms of $\mathbb{ROTang}$. This completes the proof. ☐

3 Reducing the unknotting problem to the braided monoidal word problem

So far, Corollary 25 only reduces the unknot problem to the word problem for $\mathbb{ROTang}$ while our goal is to reduce it to the word problem for braided monoidal categories. The category $\mathbb{ROTang}$ can be presented as a free braided monoidal category but that requires additional equations between the generators representing the caps and cups. In this section, we show how these equations can be eliminated too. We call *unknot diagram* any knot diagram which is isotopic to the unknot.

Definition 26. *The category* $\mathbb{CC}$ *is the free braided monoidal category generated by objects* $\{\uparrow,\downarrow\}$ *and morphisms* {  }.

It is important to note that no equations are imposed between the morphism generators, unlike in $\mathbb{ROTang}$. Therefore, there exists a functor from $\mathbb{CC}$ to $\mathbb{ROTang}$, mapping the generators of $\mathbb{CC}$ to the corresponding units and counits in $\mathbb{ROTang}$, but the reverse mapping would not be functorial.

3.1 Cap-cup cycles

In this section we introduce a more precise invariant than the turning number: the sequence of caps and cups encountered while following the strand of a knot.

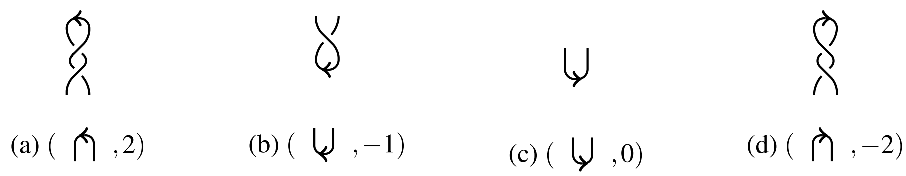


Figure 9: Examples of twisted cap-cups

Definition 27. *A **cap-cup cycle** is a finite sequence of elements in* { $\overleftarrow{\cap}$, $\overrightarrow{\cap}$, $\overleftarrow{\cup}$, $\overrightarrow{\cup}$ } *considered up to cyclic permutation in which caps and cups alternate. The turning number of a cap-cup cycle is the sum of the turning numbers of its elements, defined as in Definition 22.*

The cap-cup cycle is intended to replace the turning number in a context where eliminating caps and cups using the adjunction equations is not allowed.

Definition 28. *Given a knot $f \in \mathbb{CC}$, its cap-cup cycle* $\mathrm{ccc}(f)$ *can be obtained by starting from any strand in f, following it in the direction indicated by its type and recording all the caps and cups encountered until one travels back to the starting point. This cycle is invariant under all axioms of a braided monoidal category.*

For instance, the knot of Figure 6 has cap-cup cycle ($\overleftarrow{\cap}$, $\overrightarrow{\cup}$, $\overrightarrow{\cap}$, $\overleftarrow{\cup}$).

Lemma 29. *For all knot diagrams $f \in \mathbb{CC}$,* $\mathrm{ccc}(f)$ *is of even length, and* $T(\mathrm{ccc}(f)) = T(f)$.

Lemma 30. *For all cap-cup cycles c such that $T(c) = \pm 2$, one can construct a knot diagram $f \in \mathbb{CC}$ without any crossings, such that* $\mathrm{ccc}(f) = c$.

Proof. By induction on the length of the cycle c. If $|c| = 2$, then $c =$ ($\overrightarrow{\cap}$, $\overleftarrow{\cup}$) or $c =$ ($\overleftarrow{\cap}$, $\overrightarrow{\cup}$), both of which can be realized by the composite of both elements. If $|c| > 2$, then there is an element $x \in c$ such that $T(x) = +1$ and another element $y \in c$ with $T(y) = -1$. One can also assume that they are adjacent in c (if all elements $x \in c$ with $T(x) = +1$ are such that the elements on their left and right also have a positive turning number, then by propagating this, all elements in the cycle have a positive turning number, which is a contradiction). Consider the cycle c' obtained by removing x and y from c. By induction, construct a knot diagram $f' \in \mathbb{CC}$ such that the cap-cup cycle of f' is c'. Now, at the point where we removed x and y, we can insert in f' a zig-zag corresponding to x and y (in the order they appeared in c), which gives us the required knot. □

To generalize this lemma to knot diagrams with crossings, we introduce a new notion of cap-cup cycle where each cap or cup can carry its own writhe.

Definition 31. *The set of **twisted cap-cups** is* $\mathbb{T} :=$ { $\overleftarrow{\cap}$, $\overrightarrow{\cap}$, $\overleftarrow{\cup}$, $\overrightarrow{\cup}$ } $\times \mathbb{Z}$.

We think of a pair $(c, w) \in \mathbb{T}$ as a cap-cup c composed with braids such that the writhe of the resulting morphism is w. Figure 9 gives a few examples of twisted cap-cups.

Definition 32. *The **turning number** of a twisted cap-cup $(c,w) \in \mathbb{T}$ is defined as $T((c,w)) = (-1)^{|w|} t(c)$. The **writhe** of a twisted cap-cup is $W((c,w)) = w$. The **signature** of a twisted cap-cup is $S((c,w)) = c$ if w is even, and c with a flipped wire orientation if w is odd.*

The signature of a twisted cap-cup is essentially obtained by applying Reidemeister type I moves to the twisted cap-cup until no crossing remains. Therefore this preserves the domain and codomain of the morphism.

Definition 33. *A* ***twisted cap-cup cycle*** *is a finite sequence of twisted cap-cups up to cyclic permutation in which caps and cups alternate. The turning number of a cap-cup cycle is the sum of the turning numbers of its elements, and similarly for its writhe.*

Definition 34. *Given a twisted cap-cup cycle c we define a cap-cup cycle $U(c)$ obtained by forgetting the writhe component in each twisted cap-cup. We also define a cap-cup cycle $S(c)$ obtained by taking the signature of each twisted cap-cup in the cycle.*

Lemma 35. *Let c be a twisted cap-cup cycle such that $T(c) = \pm 2$. One can construct an unknot diagram $R(c) \in \mathbb{CC}$ such that $W(R(c)) = W(c)$ and $\mathrm{ccc}(R(c)) = U(c)$.*

Proof. First, notice that for all twisted cap-cup cycle c, $T(S(c)) = T(c)$. So if $T(c) = \pm 2$ then $T(S(c)) = \pm 2$ and we can apply Lemma 30 to $S(c)$, obtaining a morphism f such that $\mathrm{ccc}(f) = S(c)$. Now we obtain another knot diagram $R(c)$ by replacing each cap and cup of f by the twisted cap-cup in c it was generated from. This is possible because taking the signature of a twisted cap-cup preserves the domain and codomain of the corresponding morphism. We therefore obtain $W(R(c)) = W(c)$ and $\mathrm{ccc}(R(c)) = U(c)$ as required. □

Lemma 36. *Let c be a cap-cup cycle and $w \in \mathbb{Z}$ be such that $w + \frac{T(c)}{2}$ is odd. Then we can construct an unknot diagram $f(c,w) \in \mathbb{CC}$ such that $W(f(c,w)) = w$ and $\mathrm{ccc}(f(c,w)) = c$.*

Proof. We can view c as a twisted cap-cup cycle where all the writhe components are null. We will transform c to incorporate the writhe w in the writhe components of the cycle.

First, consider the case where $T(c) = \pm 2$. By assumption, w is therefore even. We can pick any element (a,b) of c and replace it by $(a, b+w)$, giving us a new twisted cap-cup cycle c'. We have $T(c') = T(c) = \pm 2$ so we can apply Lemma 35, giving the required morphism $R(c') =: f(c,w)$.

Second, if $T(c) = 0$. By assumption, w is odd. Again, take any element (a,b) in c and replace it by $(a, b+w)$. This changes the turning number of that element, negating its sign. Therefore the turning number of the new twisted cap-cup cycle is ± 2, and we are back to the previous case.

Third, if $|T(c)| > 2$. By symmetry let us assume $T(c) > 2$. We work by induction on $T(c)$. There are at least two elements of c with turning number $+1$, let them be (a,b) and (a',b'). We replace them by $(a, b+1)$ and $(a', b'-1)$ respectively. We have $t((a,b+1)) = t((a',b'-1)) = -1$ so this reduces the turning number by 4, keeps the writhe unchanged and keeps $U(c)$ unchanged. So we can obtain the required morphism by induction. □

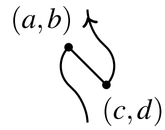
The following lemma establishes that the way the writhe is spread on the elements of a twisted cap-cup cycle does not actually matter. The writhe can be transferred between any two elements without resorting to Reidemeister I or zig-zag elimination.

Lemma 37. *Let c, c' be twisted cap-cup cycles such that $W(c) = W(c')$, $T(c) = T(c') = \pm 2$ and $U(c) = U(c')$. Then $R(c)$ is isotopic to $R(c')$ via the axioms of braided monoidal categories.*

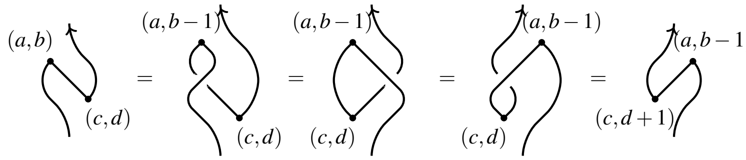
Proof. We define a relation $\diamond$ on twisted cap-cup cycles: $c \diamond c'$ when c' can be obtained from c by replacing two consecutive elements $(a,b), (c,d)$ by $(a, b-1), (c, d+1)$.

Let c, c' be twisted cap-cup cycles as in the lemma. We first show that if $c \diamond c'$ then $R(c)$ is isotopic to $R(c')$ as a braided monoidal morphism.

If $T((a,b)) = -T((c,d))$ then the sequence $(a,b),(c,d)$ is realized in $R(c)$ as follows, up to vertical and horizontal symmetries:

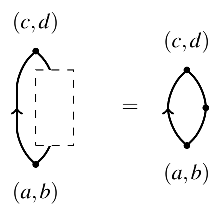


where the morphisms are composed of a single cap or cup, followed by braids to obtain the desired writhe. We have the following regular isotopy:

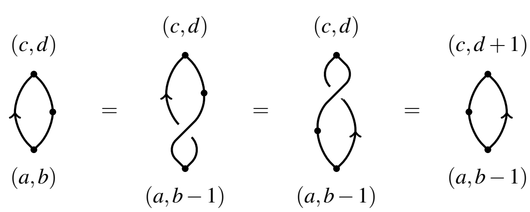


Note that the first and last equalities are not Reidemeister I moves: they can simply be expressed as unboxing the composite morphisms (a,b) and $(c,d+1)$, possibly with the help of Reidemeister II moves to create braids when required. This shows that $R(c)$ is isotopic to $R(c')$ as a braided monoidal morphism.

If $T((a,b)) = T((c,d))$ then the sequence $(a,b),(c,d)$ is realized in $R(c)$ as follows, again up to vertical and horizontal symmetries:



The dashed area in the left-hand side represents the rest of the knot. Because by construction we know that it does not cross the wire passing on its left, nor is it connected with anything else, we can abstract it away as a simple morphism taking one wire as input and one wire as output, as in the right-hand side. Then:

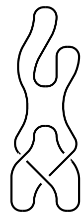


So again $R(c)$ is isotopic to $R(c')$.

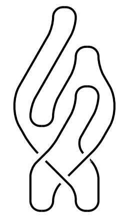
So the $\diamond$ relation respects braided monoidal isotopy. But now, by assumption $W(c) = W(c')$ and $U(c) = U(c')$. By a sequence of $\diamond$ steps one can transfer the writhe of any element of c to any other element. So c and c' are related by a series of $\diamond$ steps, so they are equal as braided monoidal morphisms. □

3.2 Bridge isotopy

In this section, we introduce a notion of knot isotopy which forbids the elimination of caps and cups, but still allows Reidemeister I moves.



(a) A knot diagram not in bridge position



(b) An equivalent diagram in bridge position

Figure 10: Bridge position

Definition 38. *A knot diagram $k \in \mathbb{CC}$ is in* ***bridge position*** *if all caps appear above of all cups in its string diagram. The number of caps (or equivalently cups) is called the* ***bridge number*** *of the diagram.*

For instance, all knot diagrams of Figure 7 are in bridge position. Figure 10 shows a knot diagram that is not in bridge position and an equivalent diagram in bridge position. The following lemma shows that any knot diagram can be put in bridge position without cancelling any zig-zag, as illustrated by Figure 10.

Lemma 39. *Any knot diagram $k \in \mathbb{CC}$ can be expressed in bridge position via the axioms of braided monoidal categories.*

Proof. While there is a cap or cup that is not on the first or last slice of the diagram, pull the cup down or pull the cap up using the pull-through move (naturality of the braid). This move can be executed regardless of the surroundings of the cap or cup. □

Note that bridge positions are not unique and there are generally multiple pull-through moves available to pull a cap or cup towards the boundary of the diagram.

Definition 40 [18, 10]. *A* ***bridge isotopy*** *between two knot diagrams in bridge position is a sequence of moves (including Reidemeister I) such that at each step the diagram is in bridge position.*

Note that because cups and caps are required to stay apart throughout the isotopy, the bridge number of the diagram is preserved by bridge isotopy.

Theorem 41 [18]. *Let K, K' be two diagrams of the unknot in bridge position, with an equal bridge number. Then they are in bridge isotopy.*

3.3 Unknotting with braided monoidal categories

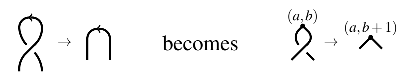
We can now combine the results above to establish a polynomial time reduction between the unknotting problem and the word problem for braided monoidal categories.

Lemma 42. *Let k be a diagram of the unknot. Then it is braided monoidal isotopic to $f(\mathrm{ccc}(k), W(k))$.*

Proof. Recall that $f(\mathrm{ccc}(k), W(k))$ is the diagram of the unknot constructed in Lemma 36 so that its cap-cup cycle and writhe match those of k.

First, the bridge number of k and $f(\mathrm{ccc}(k), W(k))$ are equal since $\mathrm{ccc}(f(\mathrm{ccc}(k), W(k))) = \mathrm{ccc}(k)$. So by Theorem 41, the two diagrams are in bridge isotopy. This is not quite enough for us since this bridge isotopy might contain Reidemeister I moves, which are not allowed in braided monoidal isotopy.

To get rid of those Reidemeister I moves, we follow the same approach as Theorem 23. First, we view all caps and cups present at all stages of the isotopy as twisted caps and cups with a null writhe component. Then, scanning the isotopy from start to end, we replace Reidemeister I moves by identities (when the Reidemeister I move cancels a braiding) or by Reidemeister II moves (when the Reidemeister I move introduces a braiding). Doing so, we bundle up the leftover braid with the cap or cup in the writhe component of the twisted cap-cup.



Since the isotopy is a bridge isotopy, caps and cups never get cancelled so adding this writhe component does not prevent any further step of the isotopy.

After this transformation, the target of the isotopy might have some additional writhe components on some caps and cups. But the original target was $f(\mathrm{ccc}(k), W(k))$, which was defined as $R(c)$, the realization of a twisted cap-cup cycle c. So the new target can also be seen as the realization of another twisted cap-cup cycle c', which has identical writhe and cap-cup cycle, because it is in braided monoidal isotopy with the source. Therefore we can apply Lemma 37 and obtain a braided monoidal isotopy between the new target of our isotopy and $f(\mathrm{ccc}(k), W(k))$, completing the proof. □

Theorem 43. *The unknotting problem can be polynomially reduced to the word problem for braided monoidal categories.*

Proof. Given a knot diagram k, we convert it to a braided monoidal word problem as follows. First, we orient it in an arbitrary way, obtaining a morphism $k' \in \mathbb{ROTang}$. We compute its writhe $W(k')$ and cap-cup cycle $\mathrm{ccc}(k')$. Then we compute $f(\mathrm{ccc}(k'), W(k'))$. All these steps can be done in polynomial time. Finally, the corresponding word problem is to determine whether k and $f(\mathrm{ccc}(k'), W(k'))$ are in braided monoidal isotopy. If they are, then k is the unknot. If they are not then by Lemma 42, k is knotted. □

A summary of the process of deciding whether a knot is unknotted given an algorithm to solve the word problem for braided monoidal categories is given in Figure 11.

Corollary 44. *The word problem for the 3-cells of free Gray categories is at least as hard as the unknotting problem.*

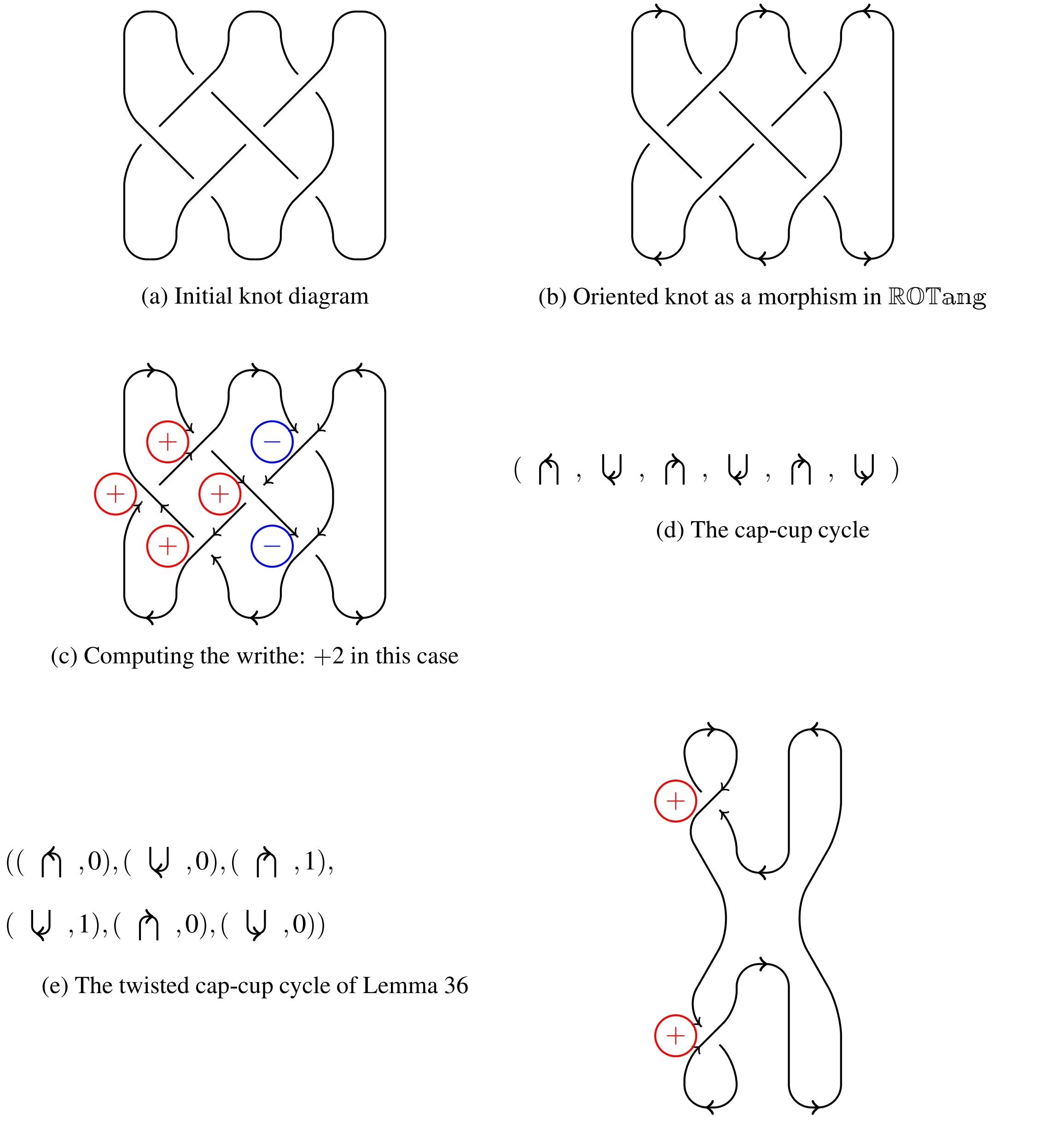
Proof. Implied by the characterization of doubly degenerate Gray categories as braided monoidal categories [8]. □

Conclusion

We have established a connection between two areas. On one side, word problems arising naturally in category theory, which have not been studied much from a computational perspective so far. On the other side, the unknotting problem, which has been studied by knot theorists for more than a century.

Our hope with this connection is to make it evident that much more work is required on word problems in category theory, especially if we hope to develop practical proof assistants for higher categories. To our knowledge, no algorithm for the braided monoidal word problem is known to date.

Conjecture 45. *The word problem for 3-cells of Gray categories (and hence cells of braided monoidal categories) is decidable.*



(a) Initial knot diagram

(b) Oriented knot as a morphism in $\mathbb{ROTang}$

(c) Computing the writhe: $+2$ in this case

(d) The cap-cup cycle

(e) The twisted cap-cup cycle of Lemma 36

(f) The realization of Lemma 35

Figure 11: Entire process of detecting knotedness using a solution to the braided monoidal word problem

Again, the word problem we mean here is deciding the equality of morphisms up to the axioms of Gray categories and nothing else. Note that the naive algorithm consisting in exploring all expressions reachable from a given expression does not terminate, since the Reidemeister II move can be applied indefinitely. This makes approaches such as that of [16] inapplicable. Furthermore it is possible that the word problem becomes undecidable at a higher level, perhaps for similar reasons that the isotopy of four-dimensional manifolds is undecidable [17, 2].

Another natural question arising from our work is whether the problem of knot equivalence could be reduced to the word problem for braided monoidal categories. Knot equivalence is the problem of determining if two knot diagrams represent the same knot. In this context, it seems more difficult to suppress the need for the yanking equations, so our results do not seem to adapt easily to this more general case.

References

[1] Emil Artin (1947): *Theory of Braids*. Ann. of Math 48(2), pp. 101–126, doi:10.2307/1969218.

[2] W. W. Boone, W. Haken & V. Poénaru (1968): *On Recursively Unsolvable Problems in Topology and Their Classification*. In H. Arnold Schmidt, K. Schütte & H. J. Thiele, editors: Studies in Logic and the Foundations of Mathematics, Contributions to Mathematical Logic 50, Elsevier, pp. 37–74, doi:10.1016/S0049-237X(08)70518-4.

[3] M. Dehn (1910): *Über Die Topologie Des Dreidimensionalen Raumes. (Mit 16 Figuren Im Text)*. Mathematische Annalen 69, pp. 137–168, doi:10.1007/BF01455155.

[4] Patrick Dehornoy (2007): *Efficient Solutions to the Braid Isotopy Problem*. arXiv:math/0703666, doi:10.1016/j.dam.2007.12.009. ArXiv:math/0703666.

[5] Antonin Delpeuch (2020): *The Word Problem for Double Categories*. Theory and Applications of Categories 35, pp. 1–18. ArXiv:1907.09927.

[6] Antonin Delpeuch & Jamie Vicary (2018): *Normalization for Planar String Diagrams and a Quadratic Equivalence Algorithm*. to appear in Logical Methods in Computer Science. ArXiv:1804.07832.

[7] Peter J Freyd & David N Yetter (1989): *Braided Compact Closed Categories with Applications to Low Dimensional Topology*. Advances in Mathematics 77(2), pp. 156–182, doi:10.1016/0001-8708(89)90018-2.

[8] Nick Gurski & Eugenia Cheng (2011): *The Periodic Table of N-Categories II: Degenerate Tricategories*. Cahiers de Topologie et Géométrie Différentielle Catégoriques 52(2), p. 45.

[9] W. Haken (1961): *Theorie Der Normalflächen*. Acta Math. 105, pp. 245–375, doi:10.1007/BF02559591.

[10] Yeonhee Jang, Tsuyoshi Kobayashi, Makoto Ozawa & Kazuto Takao (2019): *Stabilization of Bridge Decompositions of Knots and Bridge Positions of Knot Types (The Theory of Transformation Groups and Its Applications)*. RIMS Kokyuroku 2135, pp. 23–28.

[11] André Joyal & Ross Street (1986): *Braided Monoidal Categories*. Mathematics Reports 86008.

[12] André Joyal & Ross Street (1991): *The Geometry of Tensor Calculus, I*. Advances in Mathematics 88(1), pp. 55–112, doi:10.1016/0001-8708(91)90003-p.

[13] André Joyal & Ross Street (1993): *Braided Tensor Categories*. Advances in Mathematics 102(1), pp. 20–78, doi:10.1006/aima.1993.1055.

[14] Marc Lackenby (2015): *A Polynomial Upper Bound on Reidemeister Moves*. Annals of Mathematics, pp. 491–564, doi:10.4007/annals.2015.182.2.3.

[15] Marc Lackenby (2019): *The Efficient Certification of Knottedness and Thurston Norm*. arXiv:1604.00290 [math]. ArXiv:1604.00290.

[16] M. Makkai (2005): *The Word Problem for Computads*.

[17] A. Markov (1958): *The insolubility of the problem of homeomorphy*. *Doklady Akademii Nauk SSSR* 121, pp. 218–220.

[18] Jean-Pierre Otal (1982): *Présentations en ponts du nœud trivial*. *C. R. Acad. Sci., Paris, Sér. I* 294, pp. 553–556.

[19] P. Selinger (2010): *A Survey of Graphical Languages for Monoidal Categories*. In Bob Coecke, editor: *New Structures for Physics*, *Lecture Notes in Physics* 813, Springer Berlin Heidelberg, pp. 289–355, doi:10.1007/978-3-642-12821-9_4.

[20] Bruce Trace (1983): *On the Reidemeister Moves of a Classical Knot*. *Proceedings of the American Mathematical Society* 89(4), pp. 722–724, doi:10.1090/S0002-9939-1983-0719004-4.