Normal forms for planar connected string diagrams

Antonin Delpeuch* and Jamie Vicary†

May 25, 2018

Abstract

In the graphical calculus of planar string diagrams, equality is generated by the left and right exchange moves, which swaps the heights of adjacent vertices. We show that for connected diagrams the left- and right-handed exchanges each give strongly normalizing rewrite strategies. We show that these strategies terminate in $O(n^3)$ steps where n is the number of vertices. We also give an algorithm to directly construct the normal form, and hence determine isotopy, in $O(n \cdot m)$ time, where m is the number of edges.

1 Introduction

1.1 Overview

String diagrams are seeing increasingly broad application across theoretical computer science, in areas including quantum computation [1, 6, 7], natural language processing [5], interacting agents [9], circuit design [10], and rewriting [17]. In this paper we study *planar combinatorial string diagrams*—henceforth simply *diagrams*—in which 2-dimensional tiles with input and output wires are composed in the plane (see Figure 2 for an example), and prove results regarding normal forms.

The theory of diagrams is rich, because diagrams can be acted on by arbitrary recumbent[1] isotopies of the plane [12]. Here we quotient by local isotopy in the neighbourhood of each vertex, yielding a combinatorial reduction of the notion of diagram, in which the only nontrivial moves are the *right exchange* and *left exchange*, illustrated in Figure 1 in the most general case. These diagrams, as with all the diagrams we consider, are in *generic position*, with at most one vertex at each height.[2] We give a formal treatment of our notion of diagram and exchange moves in Section 2, but informally, two nodes with adjacent heights can be exchanged if and only if they have no common edges. As indicated in the diagram, we write $\to_{\mathrm{R}}$ and $\to_{\mathrm{L}}$ for the relations on diagrams given by right exchange and left exchange respectively.

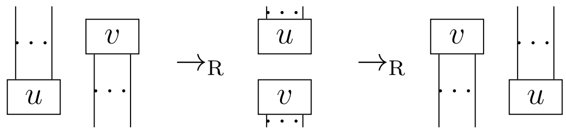
*Department of Computer Science, University of Oxford. antonin.delpeuch@cs.ox.ac.uk

†School of Computer Science, University of Birmingham and Department of Computer Science, University of Oxford. j.o.vicary@bham.ac.uk

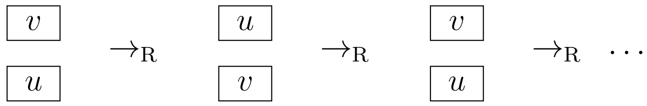
[1] An isotopy is *recumbent* when it does not cause any edge to have a point with zero tangent.

[2] This is a technical assumption that simplifies the formal development, without restricting generality, since any diagram can be made generic by an infinitesimal perturbation.

We illustrate some interesting cases of these exchange moves. In degenerate cases where u and v have no inputs or outputs, it can be possible to apply two right exchanges in sequence to the same pair of vertices:



Furthermore, if there are no edges at all, then right exchanges can be applied indefinitely:



It is well-known that if we define equality of diagrams as the least equivalence relation generated by left and right exchange, then we obtain a sound and complete graphical calculus for free bicategories generated by a signature[3], and in particular for free monoidal categories [12, 18]. An algorithm for determining equality would give a solution for the word problem in these settings. The study of right and left exchange moves on string diagrams is therefore well-motivated, but so far not well studied (although see Section 1.2.)

We contribute to this theory with the following main results. In these statements, we write n for the number of vertices in a diagram, and m for the number of edges; also, we say that a diagram is *connected* just when its graphical representation is connected in the ordinary sense.[4]

- For connected diagrams, the rewrite systems $\to_{\mathrm{R}}$ and $\to_{\mathrm{L}}$ are strongly normalizing (Theorem 16), and terminate in $O(n^3)$ steps (Theorem 19.)
- For connected diagrams, the R- or L-normal forms can be constructed in $O(n \cdot m)$ time. (Theorem 22.)

In particular, our results yield a polynomial-time algorithm for deciding equality of connected diagrams, and may have direct application in proof assistants (such as *Globular* [2, 3]) which allow direct manipulation of diagrammatic structure.

[3]Such a signature would carry type information, which we neglect here since it does not affect the geometry of the string diagram; all that matters is the length of the source and target type for each generator.

[4]While our main results concern connected diagrams, they can trivially be extended to apply to diagrams with a path from every vertex to a boundary edge, by composing with additional vertices at the top and bottom of the diagram (see Corollary 18.)

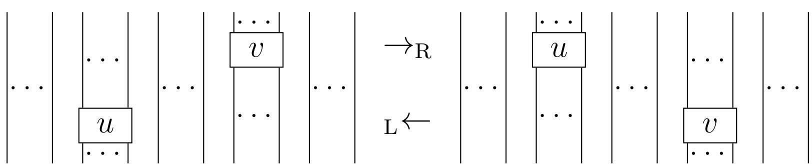


Figure 1: Right and left exchanges as rewrites on diagrams.

1.2 Related work

The use of rewriting techniques on diagrams is ubiquitous in the communities which use monoidal or higher categories, as it is much more natural than term rewriting. Diagrammatic rewriting has been studied in detail for particular signatures, such as those of boolean circuits [13, 14], the ZX-calculus [6, 7]. More generally, rewriting theory of 2-polygraphs was developed by Guiraud and Malbos [11], extending classical results on monoids. In these approaches, the goal is to decide equality of diagrams up to the axioms in the signature, and structural equalities such as the exchange law or even symmetry are strict. Our results focus instead on the structural equalities, and do not allow equalities in the signature.

The foundational work of Burroni [4] establishes the link between the word problem for an algebraic structure and the path problem in the next dimension. Makkai [16] studied the word problem for computads, in arbitrary dimension. Our work explores the computational aspects of this path problem at dimension two.

The study of equivalence in category theory often takes the form of coherence results. These state that all morphisms between given source and targets and built from a particular signature are equal. These results often rely on rewriting techniques, whose spirit was present since Mac Lane's coherence theorem for monoidal categories [15]. More recently, Forest and Mimram [8] use rewriting to prove coherence for Gray monoids. They use similar techniques, with a focus on coherence of reductions rather than their length.

1.3 Outline

This paper has the following structure. We first introduce our formalism, defining diagrams and a rewriting relation on them. In Section 3, we show that the rewriting relation terminates on connected diagrams. The asymptotic upper bound on reduction length is derived in Section 4. Section 5 shows confluence of the rewriting relation, which gives an initial algorithm to normalize diagrams. By analyzing the structure of normal forms, we describe in Section 6 a more efficient algorithm to compute them directly and hence decide the word problem.

1.4 Acknowledgements

We thank the members of the *Séminaire de catégories supérieures, polygraphes et homotopie* at IRIF, Paris for their feedback on these results, in particular Simon Forest and Samuel Mimram, and we are also grateful to Vincent Vidal for assistance with TikZ.

2 Formalism

Here we give an encoding scheme for the data of a planar combinatorial string diagram. This is essentially identical to that used by the system *Globular* [3], although the result in this section is new, and is not implied by the existing literature. This encoding scheme serves as a formal foundation for our results, although we will build most of our arguments at a more intuitive level with the corresponding graphical calculus [18]. We use this encoding to argue that certain key tests and operations can be performed in constant time.

2.1 Encoding

We begin with the formal definition of a diagram. Intuitively, a diagram comprises a number of incoming source edges, and then sequence of vertices, one at each height, each of which has some number of source and target edges.

We give an example of a diagram, together with its encoding, in Figure 2. Note that in this example diagram, and in the other diagrams later in the paper, we use small circles for the vertices, rather than the boxes which are more standard and used in Section 1.

Definition 1. For a natural number $n \in \mathbb{N}$, we define the total order $[n] = \{0, \ldots, n-1\}$.

Definition 2. A *diagram* $D = (D.S, D.N, D.H, D.I, D.O)$ comprises $D.S \in \mathbb{N}$, the number of *source edges*; $D_N \in \mathbb{N}$, the diagram *height*; and functions $D.H, D.I, D.O : [D.H] \to \mathbb{N}$ of *vertex horizontal positions*, *vertex source size* and *vertex target size* respectively.

Given a diagram, we can compute the number of edges that exist at level just below each vertex, by starting with the number of source edges $D.S$, and then supposing that each vertex $n \in [D_N]$ removes $D.I(n)$ wires and adds $D.O(n)$ wires.

Definition 3. For a diagram D, we define $D.\Delta : [D.N] \to \mathbb{N}$ as $D.\Delta(n) = D.O(n) - D.I(n)$.

Definition 4 (Wires at each level)**.** For a diagram D, we define $D.W : [D.N + 1] \to \mathbb{N}$ as $D.W(0) = D.S$, and for $n \in [D.N]$ as $D.W(n + 1) = D.W(n) + D.\Delta(n)$.

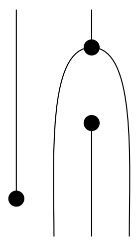
Not all diagrams will be geometrically meaningful, and we give validity conditions which check that there are enough edges available below each vertex to serve as its source edges.

Definition 5. A diagram D is *valid* when for all $n \in [D.N]$, we have $D.W(n) \geq D.H(n) + D.I(n)$.

2.2 Exchange moves

We now formalize the right and left exchange moves illustrated in Figure 1. All that needs to be checked is that there are no edges in common between the vertices.

Definition 6. For $n \in [D.N - 1]$, a diagram D *admits a right exchange move* at height n when $D.H(n + 1) \geq D.H(n) + D.O(n)$, and *admits a left exchange move* at height n when $D.H(n) \geq D.H(n + 1) + D.I(n + 1)$.



$$D = (3, 3, [0, 2, 1], [0, 1, 2], [1, 0, 1])$$

Figure 2: *Example of a diagram D together with its encoding.*

Definition 7. For a diagram D which admits a right or left exchange move at height $n \in [D_N - 1]$, its *right exchange* $D_{\mathrm{R},n}$ or *left exchange* $D_{\mathrm{L},n}$, respectively, is defined to be identical to D, except at heights n, $n+1$ as follows:

$$\begin{aligned} D_{\mathrm{R},n}.H(n) &= D.H(n+1) - D.\Delta(n) \\ D_{\mathrm{R},n}.I(n) &= D.I(n+1) \\ D_{\mathrm{R},n}.O(n) &= D.O(n+1) \\ D_{\mathrm{R},n}.H(n+1) &= D.H(n) \\ D_{\mathrm{R},n}.I(n+1) &= D.I(n) \\ D_{\mathrm{R},n}.O(n+1) &= D.O(n) \end{aligned} \qquad \begin{aligned} D_{\mathrm{L},n}.H(n) &= D.H(n+1) \\ D_{\mathrm{L},n}.I(n) &= D.I(n+1) \\ D_{\mathrm{L},n}.O(n) &= D.O(n+1) \\ D_{\mathrm{L},n}.H(n+1) &= D.H(n) + D.\Delta(n+1) \\ D_{\mathrm{L},n}.I(n+1) &= D.I(n) \\ D_{\mathrm{L},n}.O(n+1) &= D.O(n) \end{aligned}$$

Lemma 8. *For a valid diagram D which admits a right (or left) exchange move at height n, its right exchange $D_{\mathrm{R},n}$ (or left exchange $D_{\mathrm{L},n}$) is a valid diagram.*

2.3 Complexity

With respect to our data structure described in the previous subsection, it is clear that the following operations can be performed in constant time, since they involve computing fixed formulae over the natural numbers, and testing a fixed number of inequalities:

- checking whether a left or right exchange is admissible at a given height;
- given an admissible left or right exchange, computing the rewritten diagram.

We will use these as building blocks for our complexity arguments later in the paper.

3 Termination

To prove termination we first introduce the class of linear diagrams, which we will study before tackling the general case. We will see in Lemma 31 that they exhibit the longest reductions.

Definition 9. A diagram with n vertices is *linear* if its vertices form a line, i.e. k is connected to exactly $k-1$ and $k+1$ for all $1 < k < n$ (identifying its vertices with the indices $1, \dots, n$).

Definition 10. In a linear diagram of size $n \geq 2$, the final vertices are the vertices $n-1$ and n.

Definition 11. In a linear diagram, the *final interval* is the set of vertices whose height is between the height of the final vertices, including the final vertices themselves. If the final interval only consists of the final vertices, the diagram is *collapsible*.

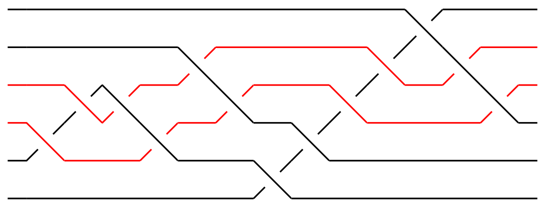
We will draw the final vertices of a linear diagram in red. Figure 3 shows examples of a linear diagram where the non-final nodes in the final interval are drawn in blue.

Definition 12. A right reduction is *collapsible* when its source and target are collapsibe, and any exchange between a non-final vertex v and a final vertex f_1 is adjacent to an exchange between v and the other final vertex f_2. In other words, all non-collapsible steps of the reduction are isolated.

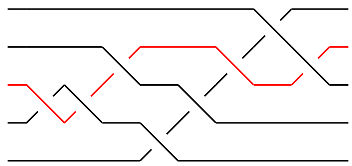
(a) A linear string diagram, not collapsible (b) A collapsible linear diagram

Figure 3: Example of linear string diagrams

We call these reductions collapsible because as the final vertices move synchronously, they can be merged together: this defines a reduction on a shorter linear diagram. Here is an example of a collapsible reduction:



By merging the final edge into a single vertex, we obtain a reduction on the induced linear diagram of smaller length:



Definition 13. A right reduction of string diagrams $r : A \to_{\mathrm{R}}^{*} B$ is called a *funnel* when:

- each non-final vertex is exchanged at most once with a final vertex.
- if an exchange involves non-final vertices u and v, then both u and v are exchanged with a final vertex in the course of the rewrite, and these two final vertices are different.

We are especially interested in the cases where the source or target of the funnel is collapsible, as in Figure 4. The name *funnel* comes from the shape of these reductions when depicted as braids: these are reductions where the final vertices converge or diverge from each other.

The following lemma decomposes reductions on linear diagrams into two parts: a collapsible part and a funnel part. This decomposition is illustrated by Figure 5. As a collapsible reduction can be seen as a reduction on a shorter linear diagram, this will let us work inductively on the length of the linear string diagram.

Lemma 14. *Let $r : A \to_{\mathrm{R}}^{*} B$ be a reduction with A collapsible. Then r can be rearranged and decomposed as*

$$c; f : A \to_{\mathrm{R}}^{*} X \to_{\mathrm{R}}^{*} B$$

with c collapsible and f a funnel.

Proof. By induction on the length of the reduction. See Appendix A for the details. □

Lemma 15. *Let $r : A \to_{\mathrm{R}}^{*} B$ be a sequence of right exchanges on a linear diagram. Then r can be extended on some side such that its domain or codomain is collapsible.*

We can now show termination of right reductions. A finer analysis of the bound obtained on the length of reductions is presented in the next section.

Theorem 16. *Right reductions are terminating on connected diagrams.*

Some diagrams that are not connected as graphs but all their vertices are connected to a boundary. Theorem 16 can be extended to these cases.

Definition 17. A diagram D is *connected via the boundary* if all vertices in D are connected to one of the two vertical boundaries of the diagram.

Corollary 18. *Right reductions terminate on diagrams that are connected via the boundary.*

4 Upper bound on reduction length

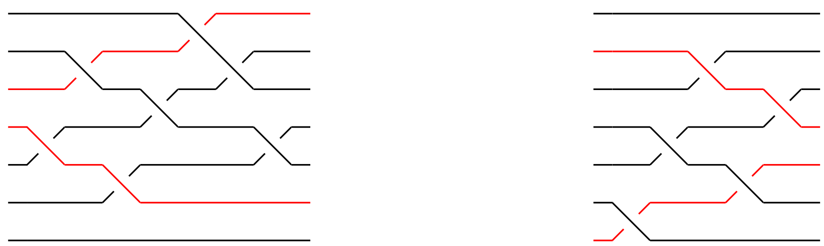
We derive a precise bound on the maximum reduction length.

Theorem 19. *Right exchanges are terminating on connected diagrams and the maximum length of a reduction on a diagram of size n vertices is $O(n^3)$. The same holds for connectedness via the boundary.*

5 Confluence

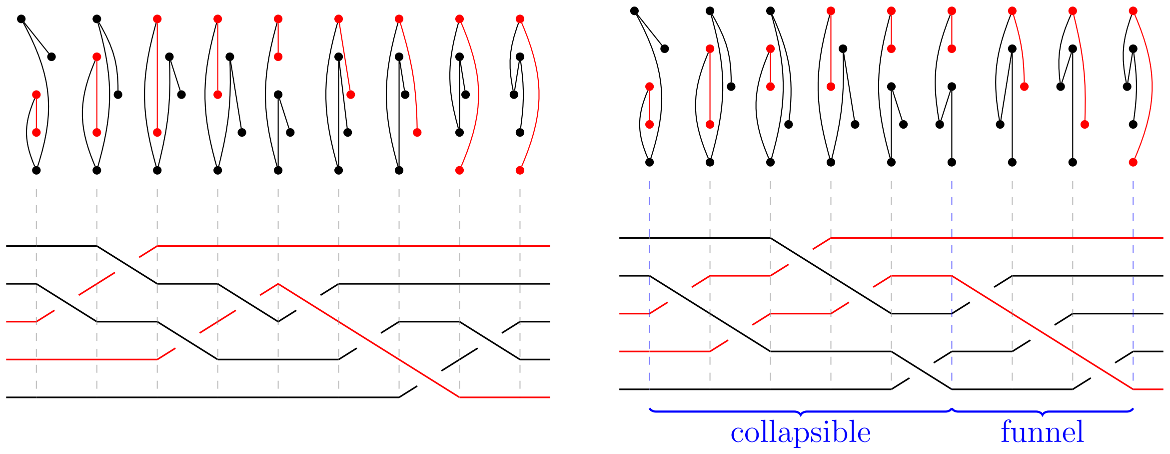
Lemma 20. *The right reduction relation is locally confluent.*

Theorem 21. *Right exchanges are confluent and therefore define normal forms for diagrams under the equivalence relation induced by exchanges. The word problem for free monoidal categories can be decided in $O(n^4)$.*



(a) A funnel with collapsible source (b) A funnel with collapsible target

Figure 4: Example of funnels



(a) Reducing a diagram to its normal form (b) Decomposition from Lemma 14

Figure 5: Decomposition into collapsible and funnel reductions

6 Computing normal forms

It follows from Theorem 21 that applying the right-exchange rewrite strategy allows us to find normal forms in $O(n^4)$ time, where n is the number of vertices. In this section we show that this complexity can be improved, giving a procedure which constructs the normal form directly in $O(nm)$ time, where n is the number of vertices and m is the number of edges.

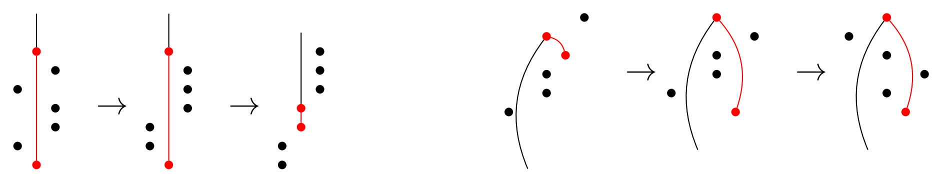
Theorem 22. *The word problem for connected string diagrams in free monoidal categories can be decided in time $O(n \cdot m)$ where n is the number of vertices and m is the number of edges.*

References

[1] Miriam Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16(9):093021, 2014. `doi:10.1088/1367-2630/16/9/093021`.

[2] Krzysztof Bar, Aleks Kissinger, and Jamie Vicary. Globular: A proof assistant for semistrict higher rewriting. In *Higher-Dimensional Rewriting and Applications (HDRA)*, 2015.

[3] Krzysztof Bar and Jamie Vicary. Data structures for quasistrict higher categories. In



(a) Appending right exchanges (b) Prepending right exchanges

Figure 6: Extending a reduction so that one end is collapsed

32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2017), 2017. `arXiv:1610.06908`, `doi:10.1109/lics.2017.8005147`.

[4] Albert Burroni. Higher-dimensional word problems with applications to equational logic. *Theoretical Computer Science*, 115(1):43–62, 1993. `doi:10.1016/0304-3975(93)90054-W`.

[5] Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. A compositional distributional model of meaning. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*, pages 133–140, 2008.

[6] Lucas Dixon, Ross Duncan, and Aleks Kissinger. Open Graphs and Computational Reasoning. *Electronic Proceedings in Theoretical Computer Science*, 26:169–180, 2010. `doi:10.4204/EPTCS.26.16`.

[7] Ross Duncan and Maxime Lucas. Verifying the Steane code with Quantomatic. *EPTCS*, 171:33–49, 2014. `doi:10.4204/EPTCS.171.4`.

[8] Simon Forest and Samuel Mimram. Coherence of Gray categories via rewriting. Submitted, 2018.

[9] Neil Ghani, Jules Hedges, Viktor Winschel, and Philipp Zahn. Compositional game theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2018)*, 2018. to appear. `arXiv:1603.04641`.

[10] Dan Ghica, Achim Jung, and Aliaume Lopez. Diagrammatic semantics for digital circuits. 2017. `arXiv:1703.10247`.

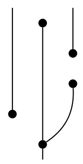
[11] Yves Guiraud and Philippe Malbos. Polygraphs of finite derivation type. *Mathematical Structures in Computer Science*, pages 1–47, 2016.

[12] André Joyal and Ross Street. The geometry of tensor calculus, I. *Advances in Mathematics*, 88(1):55–112, 1991.

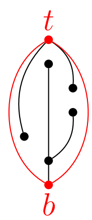
[13] Yves Lafont. Towards an algebraic theory of Boolean circuits. *Journal of Pure and Applied Algebra*, 184(2-3):257–310, 2003. `doi:10.1016/S0022-4049(03)00069-0`.

[14] Yves Lafont. Diagram rewriting and operads. *Lecture Notes from the Thematic school: Operads CIRM, Luminy (Marseille)*, 2025, 2009.

[15] Saunders Mac Lane. Natural associativity and commutativity. *Rice Univ. Studies*, 49(4):28–46, 1963.



(a) The original diagram D



(b) The transformed diagram D'

Figure 7: Adding nodes on the boundaries to make a diagram connected

[16] Michael Makkai. The word problem for computads. *Available on the author's web page http://www.math.mcgill.ca/makkai*, 2005.

[17] Samuel Mimram. Towards 3-Dimensional Rewriting Theory. *Logical Methods in Computer Science*, 10(2), 2014. `arXiv:1403.4094`, `doi:10.2168/LMCS-10(2:1)2014`.

[18] Peter Selinger. A survey of graphical languages for monoidal categories. In *New Structures for Physics*, volume 813 of *Lecture Notes in Physics*, pages 289–233. Springer, 2011.

A Material to support Lemma 14

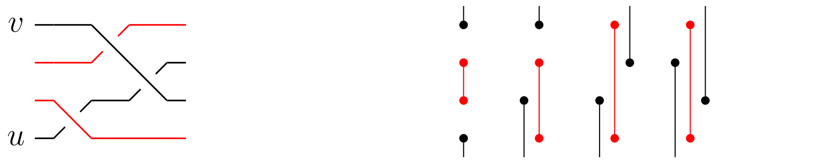
The following lemmas will establish various properties of funnels that we will need for the decomposition of Lemma 14.

Lemma 23. *Let $r : A \to_{\mathrm{R}}^{*} B$ be a funnel with A collapsible and $e : B \to_{\mathrm{R}} C$ be a right exchange of two non-final vertices u and v that are not touched by r. Then the reduction $r; e : A \to_{\mathrm{R}}^{*} B \to_{\mathrm{R}} C$ can be rearranged as $e'; r' : A \to_{\mathrm{R}} B \to_{\mathrm{R}}^{*} C$, where e' exchanges u and v in A, and r' is a funnel.*

Proof. As u and v are not touched by r, the two reductions commute directly. □

Lemma 24. *Let $r : A \to_{\mathrm{R}}^{*} B$ be a funnel reduction where A or B is collapsible. Then, the trajectory of all non-final vertices is monotone in r.*

Proof. Let us assume by symmetry that the source A of the reduction is collapsible. Consider an exchange of non-final vertices u and v in r. By definition, u and v are exchanged with two different final vertices over the course of r. Because A is collapsible, this means that both u and v have entered the final interval earlier in the reduction, by being exchanged with the bottom and top final vertices (respectively). Figure 8 shows the general position of such an exchange.



(a) The general position of an exchange of final vertices in r

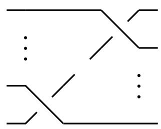
(b) Relative horizontal positions of nodes in r

Figure 8: Horizontal position of non-final nodes in a funnel

As all the exchanges involved are right exchanges, u and v are on different sides of the final edge when they are exchanged: u is on the left and v is on the right of the final edge. This means that u necessarily goes up and v goes down. As this applies to all exchanges of non-final vertices, this means that the trajectory of both vertices is monotone. □

Definition 25. An *interval right exchange* $i : A \to_{\mathrm{R}}^{*} B$ is a series of right exchanges moving a vertex x past a set of consecutive vertices $v_1, \dots, v_k$ which is adjacent to x in A and b. The vertex x is exchanged first with v_1, then v_2, up to v_k.

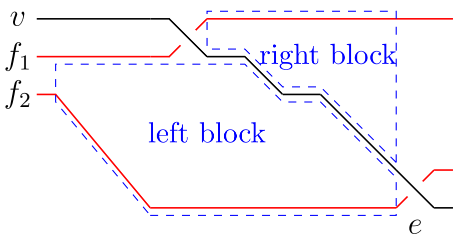
An interval right exchange looks like this:



Lemma 26. *Let $r : A \to_{\mathrm{R}}^{*} B$ be a funnel reduction with A collapsible and $e : B \to_{\mathrm{R}} C$ be an exchange of a non-final vertex v with a final vertex f_2, such that v is exchanged with the other final vertex f_1 in f. This gives a reduction path $r; e : A \to_{\mathrm{R}}^{*} B \to_{\mathrm{R}} C$. A reduction of the same length can be obtained: $i; r' : A \to_{\mathrm{R}}^{*} D \to_{\mathrm{R}}^{*} C$ where r' is final and i exchanges v with the final interval in A.*

Proof. By symmetry let us assume that f_1 is the highest final vertex, and f_2 is the lowest. Somewhere in r, v enters the final interval by being exchanged with f_1. By Lemma 24, the trajectory of v in r is monotone. In fact, because v ends up being adjacent to f_1 in B, v is exchanged exactly once with each non-final vertex that is exchanged with f_2 over the course of r.

Exchanges that do not involve v can be divided in two blocks: the ones that are on the right of the trajectory of v, and the ones that are on the left. The block on the right commutes with e because the vertices they exchange are disjoint, so we can permute the two.

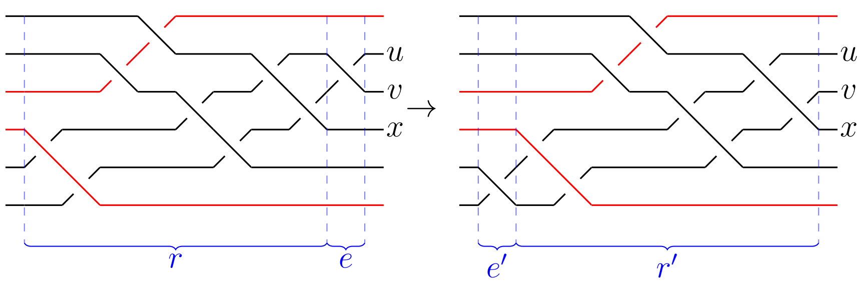


We now need to pull the block on the left through the exchanges involving v. Notice that v is the first vertex to be exchanged with f_1 over the course of r. This is because all other such vertices cannot be exchanged with v in f and v is adjacent to f_2 in B. Thus, the block on the left does not contain any exchange involving f_1: it only contains exchanges involving non-final vertices or f_2. By successive application of the Reidemeister moves, we can therefore pull the left block through the trajectory of v. $\square$

Lemma 27. *Let $r : A \to_{\mathrm{R}}^* B$ be a funnel reduction with A collapsible, followed by an exchange $e : B \to_{\mathrm{R}} B'$ of two non-final vertices u, v such that both vertices are exchanged with the same final vertex f in r. Then, the sequence $r; e$ can be rewritten as $e' : r' : A \to_{\mathrm{R}} A' \to_{\mathrm{R}}^* B'$ where e' exchanges u and v in A, and r' is a funnel.*

Proof. We show that e can be pulled through all exchanges involving u or v in r. By symmetry, we will assume that the final vertex f exchanged with u and v is the lowest one, and that u is the vertex below v in B.

By induction, consider the last exchange in r that involves one of u or v and another vertex x. Because the trajectories of u and v always go up by Lemma 24, the trajectory of x goes down. As u and v are adjacent in B, this last exchange must be between u and x, and x must have been exchanged previously with v. Moreover, this previous exchange is necessarily the last one involving v (otherwise any later exchange with y would require a later exchange between y and u). Therefore, e can be pulled through the last exchanges involving u and v.



We perform these pull-through inductively, which eventually moves e' at the beginning of the reduction. The subsequent exchange the same nodes as r in the same order, so they form a funnel. □

Proof of Lemma 14. We construct the decomposition into collapsible and funnel parts by induction on the length of the rewrite r. For length 0, the result is clear. For length 1, there are two cases: if the exchange touches a final vertex, then it goes in the funnel part of the decomposition, otherwise it forms the collapsible part.

Assume we have a rewrite of length $k+1$. Use the induction hypothesis to decompose the first k exchanges:

$$c; f; z : A \rightarrow_{\mathrm{R}}^{*} X \rightarrow_{\mathrm{R}}^{*} B' \rightarrow_{\mathrm{R}} B$$

with c collapsible and f a funnel.

If $f; z$ is also a funnel, then this gives us the required decomposition. Otherwise, this funnelity can fail for multiple reasons.

First, it can be that z exchanges a final vertex v with a non-final vertex w that is already exchanged with a final vertex in f. In this case, by Lemma 26, we can rearrange $f; z$ into $i; f'$ where f' is a funnel and i exchanges v with the final interval. As the domain of i is collapsible, i is collapsible itself so we have the required decomposition.

Second, it can be that z exchanges two non-final vertices that are not exchanged with any final vertex in f. In this case, by Lemma 23, z commutes with f: we obtain $c; z; f : A \rightarrow_{\mathrm{R}}^{*} X \rightarrow_{\mathrm{R}} X' \rightarrow_{\mathrm{R}}^{*} B$, and $c; z$ is collapsible so we have the required decomposition.

It cannot be the case that only one of the two non-final vertices z exchanges has been previously exchanged with a final vertex in f. This is because the heights of all vertices which have been exchanged with a final vertex lie in the final interval, and all other non-final vertices are outside the final interval.

Third, it can be that z exchanges two non-final vertices that are both exchanged in f with a final vertex. In this case, as we have assumed that $f; z$ is not final, it must be the vertices were exchanged with the same final vertex. We can therefore apply Lemma 27 and rearrange the rewrite into $e'; f'$ with e' exchanging the same non-final vertices as z and f' funnel. As e' is collapsible, this gives the required decomposition. □

B Further omitted material

Here we give further supporting material to the main text, including proofs of all lemmas and theorems.

Proof of Lemma 15. Our strategy to extend r depends on the topology of the final vertices. We know that vertex n is connected solely to $n-1$ and that $n-1$ is connected to both $n-2$ and n. Here are the possible ways these connections can happen:

(a) (b) (c) (d) (e) (f)

The orientation of the edges involved is preserved by the reductions so the same situation is observed in both A and B.

Consider situation (a). If the terminal layout B is not collapsible, non-final nodes are present between n and $n-1$. Some of them are on the left side of the edge connecting the final vertices and the others are on the right-hand side. Any two such nodes which are not on the same side of the final edge can be exchanged, so by appending a series of right exchanges to r we can ensure that all the ones on the left are just below $n-1$, and all the ones on the right are just above n. Then, by adding further right exchanges, we can move these non-final nodes outside the final interval, leading to a collapsible configuration. This is illustrated in Figure 6a. In the situation illustrated in Figure 6b, we choose instead to prepend right exchanges before r: this is necessary to expell vertices nested inside the cap outside the final interval. The other cases are similar: in each of them, we can either prepend or append right exchanges to obtain a collapsible configuration. □

Proof of Theorem 16. We first show termination for linear diagrams. Notice that the length of a funnel reduction on a linear diagram of length n is bounded by $O(n^2)$. This is because exchanges involving final vertices happen at most $O(n)$ times and exchanges involving only non-final vertices happen at most once per pair of non-final vertices by Lemma 24.

We can now show that right reductions terminate on linear diagrams, by induction on the length. By Lemma 15, we can assume that one end of the reduction is collapsible. By Lemma 14, we can decompose the reduction into a funnel part and a collapsible part. The collapsible part corresponds to a reduction on a smaller diagram, whose length is bounded by induction. Because an exchange involving the last vertex in the longer diagram corresponds to two exchanges in the longer diagram, we obtain a bound for the collapsible part. The funnel part is bounded as noticed above. Hence, termination holds for linear diagrams.

We now move to the general case of connected diagrams. Assume by contradiction that there is an infinite reduction on a connected diagram. By the pigeonhole principle, there is a pair of vertices that are exchanged infinitely often. Consider a simple path between these two vertices and erase all vertices not visited by this path. The infinite reduction on the connected diagram induces an infinite reduction on the linear diagram, which contradicts termination. □

Proof of Corollary 18. Let D be connected via the boundary. Consider the diagram D' obtained from D by adding two vertices b, t at the bottom and top boundaries, and adding two edges from b to t on each side of the diagram, as in Figure 7. Every edge connected to the boundary in D is connected to one of b, t in D', so D' is connected.Any right reduction on D induces a reduction of the same length on D', therefore right reductions on D' terminate. □

Definition 28. Given a reduction r on a linear string diagram of size n and an integer w, the *cost* of r at weight w is

$$\{\text{exchanges not involving vertex } n \text{ in } r\} + w \cdot \{\text{exchanges involving vertex } n \text{ in } r\}$$

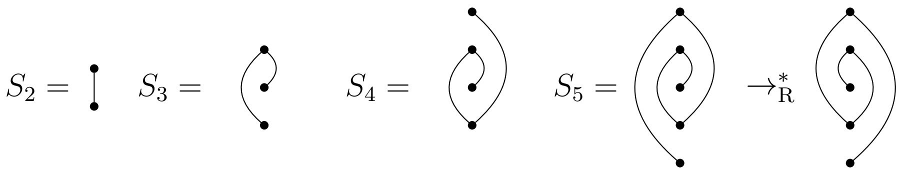
Lemma 29. *The maximum cost at weight w of a funnel with a collapsible end is $f(n, w) = O(n^2 + wn)$, where n is the length of the linear diagram.*

Proof of Lemma 29. A funnel contains two types of exchanges. Those with final vertices account for at most $n-2$ exchanges, because there is at most one for each non-final vertex. The ones with only non-final vertices are bounded by $O(n^2)$ as any pair of non-final vertices is exchanged at most once by Lemma 24. The bound follows from the definition of the cost. $\square$

Theorem 30. *The maximum cost of a right exchange on a linear diagram is $O(n^3 + w \cdot n^2)$, where n is the size of the diagram.*

Proof of Theorem 30. Let $g(n, w) = \sum_{k=1}^{n} f(k, w+n-k)$. We show that $g(n, w)$ bounds the cost of any right exchange on a linear diagram of size n. By Lemma 29, the desired bound will follow. We work induction on n. For $n \leq 1$, no right exchanges can be performed, so the bound holds. Consider a reduction $r : A \to_{\mathrm{R}}^{*} B$ on a linear diagram of size n. By Lemma 15, we can assume that A or B is collapsible (up to an extension which increases the cost of r). By Lemma 14, we can rearrange the exchanges in r to obtain a funnel and a collapsible reduction. By definition, the cost of the funnel part is bounded by $f(n, w)$. For the collapsible part, consider the reduction induced by merging the final vertices together: this gives a reduction on a diagram of size $n - 1$. Each exchange involving the last vertex in this induced reduction corresponds to an exchange of both final vertices in the original reduction, which has cost $w + 1$. Therefore, by induction, the cost of the collapsible part is bounded by $g(n - 1, w + 1)$. We therefore obtain the bound $g(n - 1, w + 1) + f(n, w) = g(n, w)$ on the cost of r at weight w. $\square$

This asymptotic bound on reduction length is attained by a class of spiral-shaped diagrams:



Lemma 31. *For all n, the diagram S_n right reduces to its normal form in $\binom{n}{3}$ steps.*

Proof of Theorem 19. Consider a connected string diagram D. Pick a spanning tree on D and let D' be the string diagram obtained from D by removing all edges which are not in the spanning tree. Any reduction on D induces a reduction of the same length on D', so it is enough to bound the length of reductions on D'.

Pick an arbitrary vertex of D' as root for the tree and consider a depth-first search of D' from that root. This defines an envelope on the tree, which can be seen as a linear diagram l if we duplicate the nodes every time they are visited (see Figure 10). The length of this diagram is linear in the number of edges in D', which is linear in the number of vertices in D'.

Any right reduction on D' translates to a right reduction on L, where exchanging vertices x and y corresponds to exchanging all the copies of x and y in the same way. The reduction on L is therefore at least as long as the reduction on d'. By Theorem 30 and because the number of vertices in L is linear in n, the reduction on L has length $O(n^3)$. This bound also applies the original reduction on D' and hence on D. The same argument as Corollary 18 extends the result to connectedness via the boundary. $\square$

Proof of Lemma 31. A reduction of S_n to its normal form starts with $n - 2$ exchanges of one end with the rest, followed by the reduction for S_{n-1} where the end weighs one more vertex. Therefore, the cost of a right reduction of S_n to its normal form is

$s(n, w) = w(n-2) + s(n-1, w+1)$. We also have $s(2, w) = 0$ for all w. From this we obtain

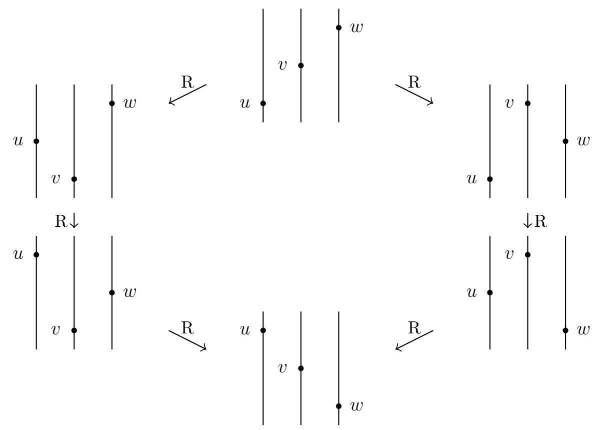
$$s(n, w) = \frac{(n-1)(n-2)(n-3+3w)}{6}$$

which gives $\binom{n}{3}$ for $w = 1$. □

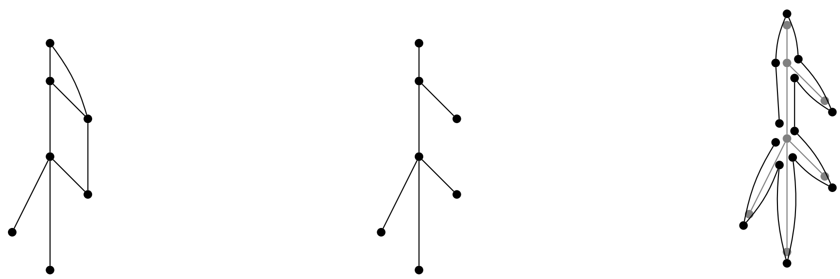
Proof of Lemma 20. Let F, G, H be diagrams in this configuration:

$$\begin{tikzcd} & F \arrow[dl, "\mathrm{R}"'] \arrow[dr, "\mathrm{R}"] & \\ G & & H \end{tikzcd}$$

If the two pairs of nodes exchanged in the two branches are disjoint, then the exchanges commute and we can close the diagram in one step: we have $H \to_{\mathrm{R}} K$ and $G \to_{\mathrm{R}} K$. Otherwise, the rewriting patterns overlap. There are nodes u, v and w in F, such that u and v are adjacent and are exchanged to obtain G, and v and w are adjacent and are exchanged to obtain H. The situation looks like this:



As u and v can be exchanged in F, there is no edge from the output of v to the input of u, and any edge going from the output of w to the input of u has to pass to the left of v. As v and w can be exchanged in F, there is no edge from the output of w to the input of v, and any edge going from the output of w to the input of u has to pass to the right of v, which is impossible by the previous observation, so there is no edge from w to u. Therefore, w and u can be exchanged both in G and H. In the resulting diagrams, we can then exchange (v, w) and (u, v) respectively, which closes the diagram. □



(a) A connected diagram d (b) A spanning tree D' on D (c) A linear diagram L obtained from D'

Figure 10: Transforming a connected diagram to a linear diagram

Proof of Theorem 21. By Theorem 19 the reduction is terminating and by Lemma 20 it is locally confluent, so by Newman's lemma right reductions are confluent. Therefore, the right normal form for a given diagram can be obtained by applying any legal right exchanges until a normal form is reached. As it is possible to find a right exchange to perform in linear time in the size of the diagram and the number of reductions is $O(n^3)$, this gives a solution in $O(n^4)$ time complexity to the word problem for free monoidal categories. □

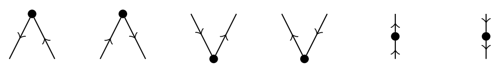
Lemma 32. *Let D be a connected diagram in right normal form. Let D' be a diagram obtained from D by adding a leaf l connected to an existing vertex v. There is a unique vertical position of l such that D' is in right normal form.*

Proof of Lemma 32. Let us first show that there is a vertical position for l such that D' is in right normal form. First, pick an initial vertical position for l, such as the position immediately above or below v (depending on the orientation of the connection between v and l). Then, normalize by applying right exchanges. All the right exchanges involve l: otherwise, by contradiction, consider the first exchange not involving l. Removing l from its domain gives us D again (because the relative positions of vertices in D has not changed), and the exchange still applies to this diagram, which contradicts normality of D. This shows the existence of the vertical position and uniqueness follows from confluence. □

This observation already gives us a way to construct the right normal form of any acyclic connected diagram. For any tree, we can remove one leaf, compute the right normal form of the remaining tree inductively, and add the leaf at the height given by the lemma. The running time of such an algorithm is quadratic, because for each vertex we are performing a linear number of exchanges which can be located efficiently (we only need to look for exchanges involving the vertex being currently added). However, this does not let us normalize cycles yet.

Definition 33. A *face* in a string diagram is a cycle whose interior region does not contain any other vertex or edge.

Definition 34. Let p be an oriented path in a diagram. For each vertex v in the interior of p, we define the number of rotations of v as follows:



$+1$ $\quad -1$ $\quad +1$ $\quad -1$ $\quad 0$ $\quad 0$

Definition 35. Given a face in a diagram D and an edge e in the face, the *mountain range* starting on e is the sequence of partial sums of number of rotations when visiting the face in direct rotation, starting from e.

Figure 11 gives an example of a mountain range for an edge in a face. Because a cycle forms a closed loop in the plane, the number of rotations of its vertices sums up to two when visited in direct rotation. This means that a mountain range always stops 2 levels higher than it started.

Definition 36. An edge in a face is *eliminable* if the mountain range starting from it never reaches 0 after the first step.

For instance, the edge above is eliminable, but its predecessor is not because the montain range starts with a valley that goes at level -1 and then 0.

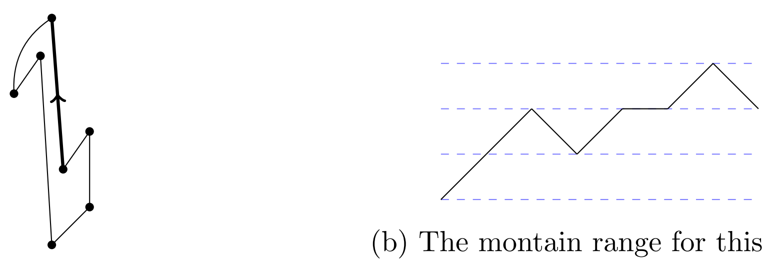
Lemma 37. *In each face there are exactly two eliminable edges.*

Proof of Lemma 37. Pick an edge in the face and draw the mountain range for it. Let m be the minimum level it reaches. Consider the last edge to reach m, we will denote it by e_1. The mountain range on the right of m never goes below $m+1$ by definition. When drawing the mountain range for e_1, the left part of the range is shifted upwards by 2, so this part never goes below 1 when drawn as part of the mountain range for e_2. So e_1 is eliminable. Similarly, consider the last edge e_2 to reach $m+1$: it is also eliminable for the same reason. These are the only two edges which satisfy the criterion. □

Lemma 38. *Let D be a connected diagram in right normal form and e be an eliminable edge in a face of D. Then the diagram D' obtained from D by removing e is in right normal form.*

Proof of Lemma 38. Consider such an edge. We first analyze what it means to be eliminable in geometrical terms. Let us call u the starting point of e and v its end point. We know that e is immediately followed by a left turn (number of rotations $+1$) at v. The next vertex where a rotation happens w also has rotation number $+1$ (otherwise the number of rotations from e to the edge after w would be null). By symmetry let us assume that e points upwards when travelling in the direct orientation on the face.

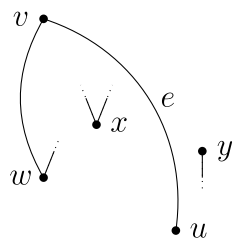
There are three sorts of right exchanges that could potentially be enabled by removing e. The first one would be exchanging the endpoints of e together, but this is impossible because of the left turn on v which imposes a horizontal ordering. The second one would be exchanging one of the endpoints of e with another vertex. This other vertex must be in the interval between the endpoints (otherwise the exchange was already possible before). That is not possible for v because of the left turn there. For u, this would require having another vertex x immediately to the left of e with no edge linked from below. We will see later that this is not possible. Finally, the third case consists in exchanging two nodes x and y between u and v, x immediately to the left of e with no edge linked from below, and y immediately to the right of e with no edge from above. We will show that no such x exists.



(a) An edge in a face

(b) The montain range for this edge

Figure 11: Example of a chosen edge in a face and its mountain range



Because e is the right boundary of the face, such an x must be a part of the boundary of the face. As part of this cycle, it has two edges coming from above. Browsing the cycle in the direct orientation can visit x in two directions: from left to right or from right to left.

If x is visited from left to right, this contradicts the fact that x is immediately to the left of e, because the interior of the face is contained between the two edges linked to x.

If x is visited from right to left, consider the path from w to x. It starts upwards and ends downwards, so it has odd number of rotation. As x itself is a right turn, this number cannot be negative: otherwise, travelling from e to the edge following x would have null or negative number of rotation, contradicting the assumption that e is eliminable. So, the path from w to x has positive number of rotation, and therefore one edge in this path is located between x and e, which contradicts the fact that x is immediately to the left of e. □

Proof of Theorem 22. We construct the right normal form of any connected string diagram by induction on the number of edges. The initial case is clear.

Given a diagram D, there are two cases. If D has a leaf, then we remove this leaf and obtain a diagram D' with one less edge that we can inductively normalize. Then, by Lemma 32, we can deduce the right normal form for D, by inserting back the leaf at the unique spot which makes the diagram normalized. Such a spot can be found in $O(n)$. If D does not have any leaf, then it has a face. In that case, by Lemma 37, there are two eliminable edges in this face. We can remove one of them and inductively normalize the resulting diagram. By Lemma 38 we can then add the edge back and obtain the normal form for D. This can also be computed in $O(n)$. We therefore obtain a normalizing algorithm in $O(n \cdot m)$. □