Structure diagrams for symmetric monoidal 3-categories: a computadic approach

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy

in the Graduate School of the Ohio State University

By

Corey Staten, M.S.

Graduate Program in Mathematics

The Ohio State University

2018

Dissertation Committee:

Niles Johnson, Adviser

Jim Fowler

Sanjeevi Krishnan

Abstract

The stable homotopy hypothesis in dimension 2 states that the homotopy theory of Picard 2-categories is equivalent to the homotopy theory of stable 2-types. Since Picard 2-categories are fully algebraic, they give us an algebraic context in which to study stable 2-types. The corresponding Picard 2-category versions of many standard constructions on stable 2-types are not yet well understood. In chapter 1, we examine symmetric monoidal bihomomorphisms $C \to \Sigma^2 B$ at the level of representing functors, where C is a Picard 1-category and B is an abelian group. In the case that C is an abelian group, our results provide an alternative proof of the known classification result for first k-invariants of Picard n-categories.

We then turn our sights on symmetric monoidal trihomomorphisms between strict symmetric monoidal 3-categories, which would be the maps representing second k-invariants. A concrete data and axioms definition of such maps does not appear in the literature. In chapter 2, we provide a computadic framework for exploring potential structure cell boundaries and axioms for this and similar definitions. Along the way, we prove decomposition results concerning representations of cells in certain types of 3-computads, which are independently of interest. We provide an implementation of these results in an open source Python library, `CatComputad` , which is discussed in chapter 3. We then use this library to propose an extension of a structure cell family which should comprise part of the definition of the desired maps.

Acknowledgments

Special thanks are due to my adviser, Niles Johnson, whose unassailable optimism and excitement about mathematics encouraged me to just keep working.

I’d also like to extend my appreciation to the other mathematics department faculty who have given me their time and energy throughout the years, including but not limited to: Jim Fowler, John Harper, Matthew Kahle, Sanjeevi Krishnan, Facundo Mémoli, and Crichton Ogle.

Warm thanks to all of my friends, who have supported me throughout the writing process.

Vita

2009	B.S. Mathematics, The Ohio State University
2011	M.S. Mathematics, The Ohio State University
2014 to Present	Graduate Teaching Associate, The Ohio State University

Fields of Study

Major Field: Mathematics

Contents

List of Figures

FIGURE PAGE

Introduction

The original motivation for this work started with the *stable homotopy hypothesis*:

stable n-types and Picard n-categories have equivalent homotopy theories

To make this more precise:

- A stable n-type is a spectrum with trivial homotopy groups except in dimensions 0 through n.
- A Picard n-category is a (weak) symmetric monoidal n-category where all cells are invertible, and each object has a weak inverse with respect to the monoidal product.

The stable homotopy hypothesis was proven in dimension 1 in [JO12], and in dimension 2 in [GJO17]. In each case, the corresponding homotopy theory of the Picard n-categories is given by categorical equivalences. Although complicated, the definition of Picard n-category (and its corresponding maps) is fully algebraic – a particular Picard n-category is fully defined by giving a concrete set of data subject to axioms. Thus, we can understand the stable homotopy hypothesis as giving us a way to present stable n-types via sufficiently complicated algebraic objects.

Given that the stable homotopy hypothesis is true in a given dimension n, we can reapproach standard constructions on stable n-types from the algebraic perspective. Our motivating example is Postnikov decompositions; given a Postnikov tower for a stable n-type, we should hope that there exists a corresponding Postnikov tower for a presenting Picard n-category and that these are related under the appropriate equivalences.

The algebraic viewpoint gives a straight-forward way to classify the potential k-invariants of a Postnikov tower when given the associated homotopy groups: we can simply unwind the algebraic definition of the maps and equivalences involved. Unfortunately, such definitions turn out to be extremely complicated. In Chapter 1, we follow this strategy to provide a characterization of maps from a Picard 1-category to a double suspension of an abelian group. This allows us to provide an alternative proof of the previously known classification of the first k-invariant of a Picard 2-category.

Attempting to extend this strategy to the second k-invariant hits a snag: the maps involved are symmetric monoidal trihomomorphisms from a Picard 1-category to the triple suspension of an abelian group. Complete algebraic definitions of symmetric monoidal trihomomorphism do not appear in the literature. In [SP09], Schommer-Pries outlines how the families of data and axioms constituting the definition of a symmetric monoidal bicategory can be considered to belong to certain combinatorially-related families; similar observations can be made about the data and axioms for a symmetric monoidal bihomomorphism and higher transfors. Thus, one approach to providing such a definition would be to extend these families upward by one dimension in the "correct" manner.

We propose to perform such extension via a diagram filling process, where axioms and structure cell families at level n are defined by finding distinct paths consisting of $n-1$ dimensional structure cells between given boundaries of dimension $n-2$. In Chapter 2, we first use the language of computads, namely strict monoidal n-computads, to formalize the basic technique of recognizing when there exists a structure cell in a given family which has source equal to a given boundary cell; this is given in Algorithm 2.3. This algorithm relies on subalgorithms for generating decompositions of n-cells as ℓ-dimensional products for $0 \leq \ell \leq n$.

To produce algorithms for generating such decompositions, we make use of the notion of n-molecule as introduced in [Mak05]; n-molecules are representations of n-cells via certain decomposition trees. Makkai provides an algorithm for determining when two *n*-molecules are equivalent; we make use of his work to provide an alternative description of equivalence checking for cells in what we call a non-degenerate strict monoidal *n*-computad. This alternative description is framed in terms of witnesses to equality made up of transposition sequences. We're then able to prove a number of structure results about *n*-molecules and their equality witnesses, culminating in Theorem 2.44, Theorem 2.65, and Theorem 2.73.

Lastly, we have created an open source Python library `CatComputad` implementing the equality checking, decomposition, and diagram filling algorithms described above; it can be found at `https://gitlab.com/coreystaten/CatComputad` . In Chapter 3, we discuss results from the use of this library, including an upward extension of an axiom family from the definition of symmetric monoidal bihomomorphism between strict symmetric monoidal 2-categories. We hope that this library will prove to be a useful tool in exploring the algebra of such diagrams.

Chapter 1

k-Invariants for Picard 1-categories

1.1 Overview

In this chapter, we give a concrete account of the symmetric monoidal bihomomorphisms from a skeletal permutative Picard 1-category (considered as a truncated symmetric monoidal 2-category) to the double suspension of an abelian group, along with their higher transfors and equivalence classes. To unpack this a bit:

- By "skeletal", we mean a category in which all isomorphic objects are equal (although not all isomorphisms between objects need be identities).
- By "permutative", we mean a symmetric monoidal category in which the associator and unitors for the monoidal product are identity transformations.
- By "double suspension of an abelian group", we mean the 2-category with one object and and one 1-cell, whose two cells are the elements of the group. This can be made into a symmetric monoidal 2-category by taking the the monoidal product on 2-cells to be the group operation.

As shown in [JO12], every Picard 1-category is equivalent as a symmetric monoidal category to a skeletal permutative one.

In the specific case that the Picard 1-category is itself an abelian group (with objects the group elements, and monoidal product on objects the group operation), these equivalence classes appear as the k-invariants of a Picard 1-category – or alternatively, as the first k-invariants for a Picard n-category. Thus, our results yield an alternative descriptive proof of the previously known classification of such k-invariants (c.f. [JO12] Proposition 2.5).

1.2 Notation

Throughout, we'll make use of the following choices:

- C is a skeletal permutative Picard 1-category with tensor denoted $\otimes$ and braiding β, considered as a symmetric monoidal bicategory.
- B is an abelian group
- $\Sigma^2 B$ is the strict Picard 2-category with one object $*$ such that $\Sigma^2 B(*, *) = \Sigma B$, also considered as a symmetric monoidal bicategory. and tensor is multiplication on 2-cells.
- 0 is the unit of B.
- 1 is the unit of $\operatorname{ob} C$ under $\otimes$.

We'll also make use of the following facts about the structure of C.

Lemma 1.1. *For $a, b \in \operatorname{ob} C$, $\mathrm{id}_a \otimes - : \operatorname{Aut}(b) \to \operatorname{Aut}(a \otimes b)$ is a group isomorphism.*

Proof. First note that $\mathrm{id}_a \otimes -$ is a homomorphism by functoriality of $\otimes$:

$$\mathrm{id}_a \otimes \mathrm{id}_b = \mathrm{id}_{a \otimes b}$$

$$\mathrm{id}_a \otimes (g \circ f) = (\mathrm{id}_a \otimes g) \circ (\mathrm{id}_a \otimes f)$$

It's an isomorphism since

$$(f \otimes \mathrm{id}_a) \otimes \mathrm{id}_{a^{-1}} = f \otimes (\mathrm{id}_a \otimes \mathrm{id}_{a^{-1}}) = f \otimes \mathrm{id}_{a \otimes a^{-1}} = f \otimes \mathrm{id}_1 = f$$

where the leftmost equality is naturality of the associator, and the rightmost is naturality of the right unitor (which are both equality transformations since C is permutative). We have that $a \otimes a^{-1} = 1$ by skeletality. □

Corollary 1.2. $\mathrm{id}_1 \otimes - : \mathrm{Aut}(a) \to \mathrm{Aut}(a)$ *is the identity operation.*

Lemma 1.3. *For $a, b \in \mathrm{ob}\, C$ and $f \in \mathrm{Aut}(a)$, there exists a unique $g \in \mathrm{Aut}(b)$ such that $f \otimes g = \mathrm{id}_{a\otimes b}$.*

Proof. Let $g = \mathrm{id}_{a^{-1}} \otimes f^{-1} \otimes \mathrm{id}_b$. Then $g^{-1} = \mathrm{id}_{a^{-1}} \otimes f \otimes \mathrm{id}_b$ by functoriality, so $\mathrm{id}_a \otimes g^{-1} = f \otimes \mathrm{id}_b$. Composing both sides with $\mathrm{id}_a \otimes g$, we have $\mathrm{id}_{a\otimes b} = f \otimes g$, again by functoriality. Such g is unique, since given h also satisfying this condition, $(f \otimes g)(f^{-1} \otimes h^{-1}) = 1_a \otimes (gh^{-1}) = 1_a \otimes 1_b$ – so by Lemma 1.1, $gh^{-1} = 1_b$. □

Lemma 1.4. *For $a \in \mathrm{ob}\, C$, $\beta_{a,1} = 1_a = \beta_{1,a}$.*

Proof. The lefthand equality follows from permutativity, and the coherence theorem for symmetric monoidal categories applied to the diagram:

$$\begin{tikzcd}
a \otimes 1 \arrow[rr, "\beta_{a,1}"] \arrow[dr, equal, "r_a"] & & 1 \otimes a \\
& a \arrow[ur, equal, "l_a"] &
\end{tikzcd}$$

The righthand equality follows from a similar diagram. □

We introduce notation to make some later sections clearer; for $a \in C$ and $f \in \mathrm{Aut}(1)$, let $(f, a) := f \otimes \mathrm{id}_a \in \mathrm{Aut}(a)$. We note that this provides a natural decomposition of the group $\cup_a \mathrm{Aut}(a)$ under $\otimes$ as the product group $\mathrm{Aut}(1) \times \mathrm{ob}\, c$. Composition makes sense since $(f \otimes \mathrm{id}_a) \otimes (g \otimes \mathrm{id}_b) = (f \otimes g) \otimes (\mathrm{id}_{a\otimes b})$ by functoriality and Lemma 1.4 – note that $f \otimes g = f \circ g$ for two elements of $\mathrm{Aut}(1)$ by the usual Eckmann-Hilton argument. Every element is represented in this manner because $- \otimes \mathrm{id}_a$ is an isomorphism from $\mathrm{Aut}(1)$ to $\mathrm{Aut}(a)$ by Lemma 1.1.

We abuse notation and let $\beta_{a,b} \in \mathrm{Aut}(1)$ denote the map $\beta_{a,b} \otimes \mathrm{id}_{a\otimes b}^{-1}$ so that we can write $(\beta_{a,b}, a \otimes b)$ wherever we would previously have written $\beta_{a,b}$. Which is being used should be clear from context. It can be verified using coherence that this β is a bihomomorphism $\mathrm{ob}\, C \times \mathrm{ob}\, C \to B$. We will often use id to indicate id_1.

1.3 Data and axioms for a symmetric monoidal bihomomorphism

An examination of the definition of symmetric monoidal bihomomorphism gives us a classification of $\mathrm{SymMonBiFun}(C, \Sigma^2 B)$ as being setwise isomorphic to the following data subject to the following constraints. A full account of the rather extensive definition (as well as the definitions of higher transfors we make use of later) can be found in [SP09], with some external references to the literature. We have maintained notation consistent with Schommer-Pries, except the he uses ϕ for both our ϕ and ψ below.

Throughout, let $a, b, c, d \in \mathrm{ob}\, C$ and f, g, h, k each lie in some automorphism group. When necessary for clarity, we will mark the automorphism group of each map using the notation f^a to mean $f \in \mathrm{Aut}(a)$.

First, the data:

- For all a, f, and g, a $\phi_{f^a, g^a} \in B$.
- For all a, a $\psi^a \in B$.
- For all a, b, f, and g, a $\chi_{f^a, g^b} \in B$
- For all a, b, and c, an $\omega_{a,b,c} \in B$
- For all a, a $\gamma_a \in B$ and a $\delta_a \in B$
- For all a and b a $u_{a,b} \in B$

And now the constraints:

F.1 $\phi_{h^a, g^a} - \phi_{h^a, (g\circ f)^a} + \phi_{(h\circ g)^a, f^a} - \phi_{g^a, f^a} = 0$

F.2 $\psi^a = -\phi_{f^a,\mathrm{id}_a} = -\phi_{\mathrm{id}_a,f^a}$

F.3 $\chi_{f^a,g^b} + \chi_{h^a,k^b} - \chi_{(h\circ f)^a,(k\circ g)^b} = \phi_{(f\otimes g)^{a\otimes b},(h\otimes k)^{a\otimes b}} - \phi_{f^a,h^a} - \phi_{g^b,k^b}$

F.4 $\chi_{\mathrm{id}_a,\mathrm{id}_b} = \psi^a + \psi^b - \psi^{a\otimes b}$

F.5 $\chi_{f^a,g^b} - \chi_{f^a,(g\otimes h)^{b\otimes c}} + \chi_{(f\otimes g)^{a\otimes b},h^c} - \chi_{g^b,h^c} = 0$

F.6 $\chi_{\mathrm{id}_1,f} = \chi_{f,\mathrm{id}_1} = 0$

F.7 $\chi_{f,g} = \chi_{g,f}$

F.8 $\chi_{\mathrm{id}_{a\otimes b\otimes c},\mathrm{id}_d} + \chi_{\mathrm{id}_a,\mathrm{id}_{b\otimes c\otimes d}} = \omega_{a,b,c} - \omega_{a,b,c\otimes d} + \omega_{a,b\otimes c,d} - \omega_{a\otimes b,c,d} + \omega_{b,c,d}$

F.9 $\omega_{a,1,b} = \delta_a + \gamma_b + 2\chi_{\mathrm{id}_a,\mathrm{id}_b}$

F.10 $\omega_{a,b,c} - \omega_{b,a,c} + \omega_{b,c,a} = u_{b\otimes c,a} - u_{b,a} - u_{c,a} - \chi_{\mathrm{id}_b,\beta_{a,c}} - \chi_{\beta_{a,b},\mathrm{id}_c}$

F.11 $-\omega_{a,b,c} + \omega_{a,c,b} - \omega_{c,a,b} = u_{c,a\otimes b} - u_{c,a} - u_{c,b} - \chi_{\beta_{a,c},\mathrm{id}_b} - \chi_{\mathrm{id}_a,\beta_{b,c}}$

F.12 $\phi_{\beta_{b,a},\beta_{a,b}} = \psi^{a\otimes b} + u_{b,a} + u_{a,b}$

Lemma 1.5. *For all $a, b \in \operatorname{ob} C$, $\chi_{\mathrm{id}_a,\mathrm{id}_b} = \omega_{1,a,1} = 0$.*

Proof. In **F.8** set $b, d = 1$:

$$\chi_{\mathrm{id}_{a\otimes c},\mathrm{id}_1} + \chi_{\mathrm{id}_a,\mathrm{id}_c} = \omega_{a,1,c} - \omega_{a,1,c} + \omega_{a,c,1} - \omega_{a,c,1} + \omega_{1,c,1}$$

which, applying **F.6** reduces to

$$\chi_{\mathrm{id}_a,\mathrm{id}_c} = \omega_{1,c,1}$$

Since this holds for any a, we have again by **F.6** that

$$\chi_{\mathrm{id}_a,\mathrm{id}_c} = \omega_{1,c,1} = \chi_{\mathrm{id}_1,\mathrm{id}_c} = 0$$

which gives the desired result after renaming variables. □

Lemma 1.6. *For all $a \in \operatorname{ob} C$, $u_{a,1} + u_{1,a} = 0$*

Proof. Notice that the left hand side of **F.11** under the substitution $a \to b, b \to c, c \to a$ is the negative of the left hand side of **F.10**. Equating right hand sides (with the negative), we get:

$$-u_{a,b\otimes c} + u_{a,b} + u_{a,c} + \chi_{\beta_{b,a},\mathrm{id}_c} + \chi_{\mathrm{id}_b,\beta_{c,a}} = u_{b\otimes c,a} - u_{b,a} - u_{c,a} - \chi_{\mathrm{id}_b,\beta_{a,c}} + -\chi_{\beta_{a,b},\mathrm{id}_c}$$

Letting $c = 1$ and applying **F.6**, we get

$$u_{a,1} + \chi_{\mathrm{id}_b,\beta_{1,a}} = -u_{1,a} - \chi_{\mathrm{id}_b,\beta_{a,1}}$$

The β's are identities by Lemma 1.4, so we can apply Lemma 1.5 to get

$$u_{a,1} + u_{1,a} = 0$$

as desired. □

Lemma 1.7. *ψ is a homomorphism.*

Proof. This is **F.4** after applying Lemma 1.5. □

Lemma 1.8. *For any $a \in \mathrm{ob}\, C$, $2\psi^a = 0$.*

Proof. Let $b = 1$ in **F.12** to get

$$\phi_{\mathrm{id}_a,\mathrm{id}_a} = \psi^a + u_{1,a} + u_{a,1}$$

Apply Lemma 1.6 and **F.2** to see that $2\psi^a = 0$. □

Lemma 1.9. $\phi_{(f,1),(g,1)} = -\chi_{(f,1),(g,1)}$

Proof. In **F.3**, set $f = k = \mathrm{id}_1$, $a = b = 1$. We get

$$\chi_{(\mathrm{id},1),(g,1)} + \chi_{(h,1),(\mathrm{id},1)} - \chi_{(h,1),(g,1)} = \phi_{(g,1),(h,1)} - \phi_{(\mathrm{id},1),(h,1)} - \phi_{(g,1),(\mathrm{id},1)}$$

By **F.2** and Lemma 1.8, the righthand two terms sum to 0. By **F.6**, the lefthand two terms are both 0. Applying **F.7**, we have

$$\phi_{(g,1),(h,1)} = -\chi_{(g,1),(h,1)}$$

□

Lemma 1.10. $\delta_1 + \gamma_1 = 0$

Proof. By Lemma 1.5, $\omega_{1,1,1} = 0$. Now **F.9** with $a = b = 1$ says that $\omega_{1,1,1} = \delta_1 + \gamma_1$, giving the result. □

The following are useful later for proving Lemma 1.20. First, we can always rewrite ϕ in terms of ψ and χ:

Lemma 1.11. $\phi_{(f,a),(g,a)} = \chi_{(f,1),(\mathrm{id},a)} + \chi_{(g,1),(\mathrm{id},a)} - \chi_{(f,1),(g,1)} - \chi_{(f\circ g,1),(\mathrm{id},a)} - \psi^a$

Proof. Set $g = k = (\mathrm{id}, b)$, $f = (f, a)$, $h = (h, a)$ in **F.3** and apply **F.2** to get

$$\phi_{(f,a\otimes b),(h,a\otimes b)} - \phi_{(f,a),(h,a)} = \chi_{(f,a),(\mathrm{id},b)} + \chi_{(h,a),(\mathrm{id},b)} - \psi^b - \chi_{(f\circ h,a),(\mathrm{id},b)}$$

Set $a = 1$ in the above to get

$$\phi_{(f,b),(h,b)} - \phi_{(f,1),(h,1)} = \chi_{(f,1),(\mathrm{id},b)} + \chi_{(h,1),(\mathrm{id},b)} - \psi^b - \chi_{(f\circ h,1),(\mathrm{id},b)}$$

Now apply Lemma 1.9 and substitute $b \to a$, $h \to g$ to get the result. □

We can also rewrite $\chi_{(f,a),(g,b)}$ in terms of terms having either one map equal to id, or both objects equal to 1:

Lemma 1.12. $\chi_{(f,a),(g,b)} = \chi_{(\mathrm{id},a),(g,b)} + \chi_{(f,a),(\mathrm{id},b)} - \chi_{(g,1),(\mathrm{id},a\otimes b)} - \chi_{(f,1),(\mathrm{id},a\otimes b)} + \chi_{(f,1),(g,1)} + \chi_{(g\circ f,1),(\mathrm{id},a\otimes b)}$

Proof. First, we set $f = (\mathrm{id}, a)$, $k = (\mathrm{id}, b)$, $g = (g, b)$, $h = (h, a)$ in **F.3**:

$$\chi_{(\mathrm{id},a),(g,b)} + \chi_{(h,a),(\mathrm{id},b)} - \chi_{(h,a),(g,b)} = \phi_{(g,a\otimes b),(h,a\otimes b)} - \phi_{(\mathrm{id},a),(h,a)} - \phi_{(g,b),(\mathrm{id},b)}$$

Now we use **F.2** to rewrite the two relevant instances of ϕ, and Lemma 1.11 to rewrite the remaining instance:

$$\begin{aligned}
&\chi_{(\mathrm{id},a),(g,b)} + \chi_{(h,a),(\mathrm{id},b)} - \chi_{(h,a),(g,b)} = \\
&\qquad \chi_{(g,1),(\mathrm{id},a\otimes b)} + \chi_{(h,1),(\mathrm{id},a\otimes b)} - \chi_{(g,1),(h,1)} - \chi_{(g\circ h,1),(\mathrm{id},a\otimes b)} \\
&\qquad - \psi^{a\otimes b} + \psi^a + \psi 6b
\end{aligned}$$

Notice that the ψ terms cancel since ψ is a homomorphism by Lemma 1.7. After moving terms around, applying symmetry of χ and relabeling $h \to f$, this gives the result. □

Lemma 1.13. $\chi_{(f,a),(\mathrm{id},b)} = \chi_{(f,1),(\mathrm{id},a\otimes b)} - \chi_{(f,1),(\mathrm{id},a)}$

Proof. Set $f = (f,1), g = (\mathrm{id},b), h = (\mathrm{id},c)$ in **F.5** to get

$$\chi_{(f,1),(\mathrm{id},b)} - \chi_{(f,1),(\mathrm{id},b\otimes c)} + \chi_{(f,b),(\mathrm{id},c)} - \chi_{(\mathrm{id},b),(\mathrm{id},c)} = 0$$

The last term is 0 by Lemma 1.5, so relabeling $b \to a, c \to b$ gives

$$\chi_{(f,a),(\mathrm{id},b)} = \chi_{(f,1),(\mathrm{id},a\otimes b)} - \chi_{(f,1),(\mathrm{id},a)}$$

as desired. □

This allows us to give a better form of $\chi_{(f,a),(g,b)}$:

Lemma 1.14. $\chi_{(f,a),(g,b)} = \chi_{(f,1),(g,1)} + \chi_{(f\circ g,1),(\mathrm{id},a\otimes b)} - \chi_{(f,1),(\mathrm{id},a)} - \chi_{(g,1),(\mathrm{id},b)}$

Proof. Use Lemma 1.13 to rewrite the relevant terms in Lemma 1.12. □

This decomposition of χ is essential to our main theorems. It says that χ can be fully (not uniquely) defined using two functions $\chi^1 : \mathrm{Aut}(1) \times \mathrm{Aut}(1) \to B$ and $\chi^2 : \mathrm{Aut}(1) \times \mathrm{ob}\, C \to B$. Our later definition of nice bihomomorphism will be one for which $\chi^2 = 0$.

Let $T = \cup_a \mathrm{Aut}(a) \cong \mathrm{Aut}(1) \times \mathrm{ob}\, C$. There's a projection map $p : T \to \mathrm{Aut}(1)$ (namely $f^a \mapsto f^a \otimes \mathrm{id}_{a^{-1}}$) and an injection map $i : \mathrm{Aut}(1) \to T$ sending f to $(f,1)$. χ is a 2-cocycle $T \times T \to B$. The above result can be rewritten to read $\chi - \chi \circ (ip, ip) = \partial \chi^2$.

1.4 Data and axioms for a transformation

We now consider a transformation between two symmetric monoidal bihomomorphisms defined by the above data. If $\alpha : H \to \widetilde{H}$ is the transformation, and we use the notation $\widetilde{\chi}$

to denote the version of χ associated to the bihomomorphism $\widetilde{H}$, then this transformation is given by the following data:

- For all $a \in \operatorname{ob} C$ and $f \in \operatorname{Aut}(a)$, a $\theta_f \in B$.
- For all $a, b \in \operatorname{ob} C$, a $\Pi_{a,b} \in B$.
- An $M \in B$.

Subject to constraints:

T.1 $\widetilde{\chi}_{f,g} - \chi_{f,g} = \theta_{f\otimes g} - \theta_f - \theta_g$

T.2 $\theta_{\mathrm{id}_1} = 0$

T.3 $\Pi_{a,b} - \Pi_{a,b\otimes c} + \Pi_{a\otimes b,c} - \Pi_{b,c} + \theta_{\mathrm{id}_{a\otimes b\otimes c}} = -\widetilde{\omega}_{a,b,c} + \omega_{a,b,c}$

T.4 $\Pi_{1,a} + \theta_{\mathrm{id}_a} + \gamma_a - \widetilde{\gamma}_a = M = \Pi_{a,1} - \theta_{\mathrm{id}_a} + \widetilde{\delta}_a - \delta_a$

T.5 $\Pi_{a,b} - \Pi_{b,a} + \theta_{\beta_{a,b}} = \widetilde{u}_{b,a} - u_{b,a}$

Again, we prove some relevant lemmas before moving on.

Lemma 1.15. *For any $a \in \operatorname{ob} C$, $\theta_{\mathrm{id}_a} = 0$*

Proof. Add three copies of **T.3** with variable substitutions and negatives as follows:

$$\begin{aligned}
\Pi_{a,b} - \Pi_{a,b\otimes c} + \Pi_{a\otimes b,c} - \Pi_{b,c} + \theta_{\mathrm{id}_{a\otimes b\otimes c}} &= -\widetilde{\omega}_{a,b,c} + \omega_{a,b,c} \\
-\Pi_{b,a} + \Pi_{b,a\otimes c} - \Pi_{a\otimes b,c} + \Pi_{a,c} - \theta_{\mathrm{id}_{a\otimes b\otimes c}} &= \widetilde{\omega}_{b,a,c} - \omega_{b,a,c} \\
\Pi_{b,c} - \Pi_{b,a\otimes c} + \Pi_{b\otimes c,a} - \Pi_{c,a} + \theta_{\mathrm{id}_{a\otimes b\otimes c}} &= -\widetilde{\omega}_{b,c,a} + \omega_{b,c,a}
\end{aligned}$$

After cancelling terms, we end up with:

$$\theta_{\mathrm{id}_{a\otimes b\otimes c}} + (\Pi_{a,b} - \Pi_{b,a}) + (\Pi_{a,c} - \Pi_{c,a}) + (\Pi_{b\otimes c,a} + -\Pi_{a,b\otimes c}) =$$
$$\omega_{a,b,c} - \omega_{b,a,c} + \omega_{b,c,a} - (\widetilde{\omega}_{a,b,c} - \widetilde{\omega}_{b,a,c} + \widetilde{\omega}_{b,c,a})$$

Apply **T.5** and **F.10** to get

$$\theta_{\mathrm{id}_{a\otimes b\otimes c}} + \widetilde{u}_{b,a} - u_{b,a} - \theta_{\beta_{a,b}} + \widetilde{u}_{c,a} - u_{c,a} - \theta_{\beta_{a,c}} + \widetilde{u}_{a,b\otimes c} - u_{a,b\otimes c} - \theta_{\beta_{b\otimes c,a}} =$$
$$u_{b\otimes c,a} - u_{b,a} - u_{c,a} - \chi_{\mathrm{id}_b,\beta_{a,c}} + -\chi_{\beta_{a,b},\mathrm{id}_c} + \widetilde{u}_{b\otimes c,a} - \widetilde{u}_{b,a} - \widetilde{u}_{c,a} - \widetilde{\chi}_{\mathrm{id}_b,\beta_{a,c}} + -\widetilde{\chi}_{\beta_{a,b},\mathrm{id}_c}$$

Set $a, b = 1$ and use the fact that $\beta_{c,1} = \mathrm{id}_c = \beta_{1,c}$ to get (after cancelling)

$$\theta_{\mathrm{id}_c} + \widetilde{u}_{1,1} - \theta_{\mathrm{id}_1} + \widetilde{u}_{c,1} - \theta_{\mathrm{id}_c} + \widetilde{u}_{1,c} - u_{1,c} - \theta_{\mathrm{id}_c} = u_{c,1} - \chi_{\mathrm{id}_1,\mathrm{id}_a} + -\chi_{\mathrm{id}_1,\mathrm{id}_c} - \widetilde{u}_{1,1} - \widetilde{\chi}_{\mathrm{id}_1,\mathrm{id}_c} + -\widetilde{\chi}_{\mathrm{id}_1,\mathrm{id}_c}$$

Now apply **F.6**, Lemma 1.6, and **T.2** to eliminate terms and get that $\theta_{\mathrm{id}_c} = 0$ as desired.

□

1.5 Data and axioms for an equivalence

As our final layer, we consider the data associated to an equivalence. Given α as above, an equivalence additionally involves a transformation $\overline{\alpha} : \widetilde{H} \to H$ along with invertible modifications $\mathcal{E} : \overline{\alpha} \circ \alpha \to I$ and $\mathcal{F} : I \to \alpha \circ \overline{\alpha}$. The data for $\overline{\alpha}$ is of the same form as that for α, with the roles of f and $\widetilde{f}$ switched for each relevant piece of functorial data f; we denote this data by (for example) $\overline{\Pi}$ instead of Π.

The data for $\mathcal{E}$ and $\mathcal{F}$ are:

- $\forall a \in \mathrm{ob}\,\mathcal{C}$, $E_a \in B$ and $F_a \in B$.

Subject to constraints:

M.1 $\theta_f = -\overline{\theta}_f$

M.2 $E_{a\otimes b} - E_a - E_b = -\Pi_{a,b} - \overline{\Pi}_{a,b}$

M.3 $E_0 = -M - \overline{M}$

M.4 $F_{a\otimes b} - F_a - F_b = \Pi_{a,b} + \overline{\Pi}_{a,b}$

M.5 $F_0 = M + \overline{M}$

1.6 Symmetric monoidal bihomomorphisms in nice form in case $\beta = \mathrm{id}$

In the case that $\beta = \mathrm{id}$, we can show that every symmetric monoidal bihomomorphism is equivalent to one of a nice form. Specifically we will show that our given bihomomorphism H with data as above is equivalent to a bihomomorphism $\widetilde{H}$ where:

- $\widetilde{\chi}_{(f,a),(g,b)} = \widetilde{\chi}_{(f,1),(g,1)}$
- $\widetilde{\gamma}_a = \widetilde{\delta}_a = 0$
- $\widetilde{\psi}^a = 0$
- $\widetilde{\phi}_{(f,a),(g,a)} = -\widetilde{\chi}_{(f,1),(g,1)}$

Particularly, we can make the following choices for $\widetilde{H}$ given H:

- $\widetilde{\chi}_{(f,a),(g,b)} = \chi_{(f,1),(g,1)}$
- $\widetilde{\phi}_{(f,a),(g,a)} = -\chi_{(f,1),(g,1)}$
- $\widetilde{u}_{a,b} = u_{a,b} - \delta_a - \gamma_a + \delta_b + \gamma_b$
- $\widetilde{\psi}^a = 0$
- $\widetilde{\omega}_{a,b,c} = \omega_{a,b,c} - \delta_{a\otimes b} + \gamma_b + \delta_b - \gamma_{b\otimes c}$
- $\widetilde{\delta} = 0$
- $\widetilde{\gamma} = 0$

Our natural transformation α can then be chosen to be

- $\theta_{(f,a)} = -\chi_{(f,1),(1,a)}$
- $\Pi_{a,b} = \delta_a - \gamma_b$

- $M = \delta_0 (= -\gamma_0)$

$\overline{\alpha}$ is chosen to have the same data except with negative signs, and we set $E, F = 0$ for the equivalence. We must verify that this choice satisfies all of the above constraints. First, we verify that $\widetilde{H}$ is in fact an appropriate symmetric monoidal bihomomorphism.

- **F.1** follows from **F.5** for H.
- **F.2** follows from **F.6** for H.
- **F.3** follows from the same for H with $a = b = 1$, using Lemma 1.9.
- **F.4** is trivial
- **F.5** follows from the same for H with $a = b = 1$.
- **F.6** follows from the same for H.
- **F.7** follows from the same for H.
- **F.8** is Lemma 1.16 below.
- **F.9** follows directly from Lemma 1.10.
- **F.10** is Lemma 1.17 below.
- **F.11** is Lemma 1.18 below.
- **F.12** is Lemma 1.19 below.

Lemma 1.16. $\widetilde{\chi}_{\mathrm{id}_{a\otimes b\otimes c},\mathrm{id}_d} + \widetilde{\chi}_{\mathrm{id}_a,\mathrm{id}_{b\otimes c\otimes d}} = \widetilde{\omega}_{a,b,c} - \widetilde{\omega}_{a,b,c\otimes d} + \widetilde{\omega}_{a,b\otimes c,d} - \widetilde{\omega}_{a\otimes b,c,d} + \widetilde{\omega}_{b,c,d}$

Proof. Firstly, **F.6** for H eliminates the left hand side. So we need to verify

$$0 = \widetilde{\omega}_{a,b,c} - \widetilde{\omega}_{a,b,c\otimes d} + \widetilde{\omega}_{a,b\otimes c,d} - \widetilde{\omega}_{a\otimes b,c,d} + \widetilde{\omega}_{b,c,d}$$

Substituting, the sum on the right hand side becomes:

$$\begin{aligned}
&\omega_{a,b,c} - \delta_{a\otimes b} + \gamma_b + \delta_b - \gamma_{b\otimes c}\\
&- \omega_{a,b,c\otimes d} + \delta_{a\otimes b} - \gamma_b - \delta_b + \gamma_{b\otimes c\otimes d}\\
&+ \omega_{a,b\otimes c,d} - \delta_{a\otimes b\otimes c} + \gamma_{b\otimes c} + \delta_{b\otimes c} - \gamma_{b\otimes c\otimes d}\\
&- \omega_{a\otimes b,c,d} + \delta_{a\otimes b\otimes c} - \gamma_c - \delta_c + \gamma_{c\otimes d}\\
&+ \omega_{b,c,d} - \delta_{b\otimes c} + \gamma_c + \delta_c - \gamma_{c\otimes d}
\end{aligned}$$

The γ and δ terms cancel exactly, so we just need to verify that

$$0 = \omega_{a,b,c} - \omega_{a,b,c\otimes d} + \omega_{a,b\otimes c,d} - \omega_{a\otimes b,c,d} + \omega_{b,c,d}$$

But this is just **F.8** for H using Lemma 1.5 to eliminate its left hand side. □

Lemma 1.17. $\widetilde{\omega}_{a,b,c} - \widetilde{\omega}_{b,a,c} + \widetilde{\omega}_{b,c,a} = \widetilde{u}_{b\otimes c,a} - \widetilde{u}_{b,a} - \widetilde{u}_{c,a} - \widetilde{\chi}_{\mathrm{id}_b,\beta_{a,c}} - \widetilde{\chi}_{\beta_{a,b},\mathrm{id}_c}$

Proof. Since $\beta = \mathrm{id}$, the $\widetilde{\chi}$ terms are 0 by Lemma 1.5. Working out the remaining substitutions, we need to show:

$$\begin{aligned}
&\omega_{a,b,c} - \delta_{a\otimes b} + \gamma_b + \delta_b - \gamma_{b\otimes c}\\
&- \omega_{b,a,c} + \delta_{a\otimes b} - \gamma_a - \delta_a + \gamma_{a\otimes c}\\
&\omega_{b,c,a} - \delta_{b\otimes c} + \gamma_c + \delta_c - \gamma_{a\otimes c}\\
&=\\
&u_{b\otimes c,a} - \delta_{b\otimes c} - \gamma_{b\otimes c} + \delta_a + \gamma_a\\
&- u_{b,a} + \delta_b + \gamma_b - \delta_a - \gamma_a\\
&- u_{c,a} + \delta_c + \gamma_c - \delta_a - \gamma_a
\end{aligned}$$

As in the previous lemma, all γ and δ terms cancel; this reduces to **F.10** for H. □

Lemma 1.18. $-\widetilde{\omega}_{a,b,c} + \widetilde{\omega}_{a,c,b} - \widetilde{\omega}_{c,a,b} = \widetilde{u}_{c,a\otimes b} - \widetilde{u}_{c,a} - \widetilde{u}_{c,b} - \widetilde{\chi}_{\beta_{a,c},\mathrm{id}_b} - \widetilde{\chi}_{\mathrm{id}_a,\beta_{b,c}}$

Proof. The proof is analagous to that of the last lemma. □

Lemma 1.19. $\widetilde{\phi}_{\beta_{b,a},\beta_{a,b}} = \widetilde{\psi}^{a\otimes b} + \widetilde{u}_{b,a} + \widetilde{u}_{a,b}$

Proof. Since $\beta = \mathrm{id}$, both $\widetilde{\phi}$ and $\widetilde{\psi}$ are 0. Substituting, we just need to show

$$0 = u_{b,a} - \delta_b - \gamma_b + \delta_a + \gamma_a + u_{a,b} - \delta_a - \gamma_a + \delta_b + \gamma_b$$

Cancelling terms, this is just $u_{b,a} + u_{a,b} = 0$. But this follows from **F.12** for H, along with **F.2** and the fact that $\beta = \mathrm{id}$. □

Next we verify that α is a valid transformation.

- **T.1** is Lemma 1.20 below.
- **T.2** follows from **F.2** for H.
- **T.3** is Lemma 1.21 below.
- **T.4** is Lemma 1.22 below.
- **T.5** is Lemma 1.23 below.

Lemma 1.20. $\widetilde{\chi}_{f,g} - \chi_{f,g} = \theta_{f\otimes g} - \theta_f - \theta_g$

Proof. To clarify, let's switch notation in the desired result, substituting (f,a) for f and (g,b) for g:

$$\widetilde{\chi}_{(f,a),(g,b)} - \chi_{(f,a),(g,b)} = \theta_{(f\circ g,a\otimes b)} - \theta_{(f,a)} - \theta_{(g,b)}$$

Substituting definitions, it is sufficient to verify:

$$\chi_{(f,1),(g,1)} - \chi_{(f,a),(g,b)} = -\chi_{(f\circ g,1),(\mathrm{id},a\otimes b)} + \chi_{(f,1),(\mathrm{id},a)} + \chi_{(g,1),(\mathrm{id},b)}$$

Moving terms around, this is just Lemma 1.14 which is already known to hold since H is a well-defined symmetric monoidal bihomomorphism. □

Lemma 1.21. $\Pi_{a,b} - \Pi_{a,b\otimes c} + \Pi_{a\otimes b,c} - \Pi_{b,c} + \theta_{\mathrm{id}_{a\otimes b\otimes c}} = -\widetilde{\omega}_{a,b,c} + \omega_{a,b,c}$

Proof. Substituting, we can verify the target equation by verifying:

$$\delta_a - \gamma_b - \delta_a + \gamma_{b\otimes c} + \delta_{a\otimes b} - \gamma_c - \delta_b + \gamma_c + \chi_{(\mathrm{id},1),(\mathrm{id},a\otimes b\otimes c)} =$$
$$- \omega_{a,b,c} + \delta_{a\otimes b} - \gamma_b - \delta_b + \gamma_{b\otimes c} + \omega_{a,b,c}$$

But all terms cancel perfectly or are 0, so the desired equation holds. □

Lemma 1.22. $\Pi_{1,a} + \theta_{\mathrm{id}_a} + \gamma_a - \widetilde{\gamma}_a = M = \Pi_{a,1} - \theta_{\mathrm{id}_a} + \widetilde{\delta}_a - \delta_a$

Proof. Substituting, we can verify the target equation by verifying:

$$= \delta_1 - \gamma_a - \chi_{(\mathrm{id},1),(\mathrm{id},a)} + \gamma_a = \delta_1 = \delta_a - \gamma_1 + \chi_{(\mathrm{id},1),(\mathrm{id},a)} - \delta_a$$

Which is clearly true after cancelling by **F.6** and Lemma 1.10. □

Lemma 1.23. $\Pi_{a,b} - \Pi_{b,a} + \theta_{\beta_{a,b}} = \widetilde{u}_{b,a} - u_{b,a}$

Proof. Note that we're assuming $\beta_{a,b} = \mathrm{id}_{a\otimes b}$. Substituting, we can verify the target equation by verifying:

$$\delta_a - \gamma_b - \delta_b + \gamma_a - \chi_{(\mathrm{id},1),(\mathrm{id},a\otimes b)} = u_{b,a} - \delta_b - \gamma_b + \delta_a + \gamma_a - u_{b,a}$$

which reduces to $0 = 0$ after cancelling terms. □

Following the same outline, we can also prove that $\overline{\alpha}$ is well defined. There's an easy way to see that this works: for each data function μ defined for α or ν defined for H (such as $\mu = \Pi$ or $\nu = \chi$), $\overline{\mu} = -\mu$ and $\widetilde{\nu}$ switches roles with ν in all constraints (since the direction of the transformation is reversed). The combined effect is that the constraint equations for $\overline{\alpha}$ are the same as those for α, but with a negative sign applied throughout.

Lastly, we need to verify that our choice of E and F leads to a well-defined equivalence; however **M.1** through **M.5** all hold trivially from the definitions.

1.7 Form for transformations between nice form symmetric monoidal bihomomorphisms for $\beta = \mathrm{id}$

We can classify the transformations between symmetric monoidal bihomomorphisms in the nice form from the previous section. To recall, this is functors defined by data such that:

- $\chi_{(f,a),(g,b)} = \chi_{(f,1),(g,1)}$
- $\gamma_a = \delta_a = 0$
- $\psi^a = 0$
- $\phi_{(f,a),(g,a)} = -\chi_{(f,1),(g,1)}$

We can summarize the data and constraints for such a symmetric monoidal bihomomorphism as follows. It is uniquely determined by:

- A normalized symmetric 2-cocycle χ with domain $\mathrm{Aut}(1)$, subject to the constraint that $\chi_{f,g} + \chi_{h,k} - \chi_{h \circ f, k \circ g} = \chi_{f,h} + \chi_{g,k} - \chi_{f \circ g, h \circ k}$ (**F.3**, **F.5**, **F.6**, **F.7**)
- A normalized symmetric 3-cocycle (ω, u) with domain $\mathrm{ob}\, C$ (**F.8**, **F.9**, **F.10**, **F.12**)

Note that since $\beta = \mathrm{id}$, $\phi = -\chi$, and $\psi = 0$, **F.1**, **F.2** and **F.4** are automatically satisfied and add no additional constraint. In this case, **F.11** is equivalent to **F.10**, so also adds no additional constraint.

A transformation between such nice symmetric monoidal bihomomorphisms is determined by:

- A cohomology between χ and $\widetilde{\chi}$, given by θ. (**T.1**)
- A cohomology between (ω, u) and $(\widetilde{\omega}, \widetilde{u})$, given by $\Pi - M$, plus a constant $M \in B$. (**T.3**, **T.4**, **T.5**)

Note that T.2 is already implied by θ being a cohomology and adds no additional constraint.

Notice that if we have a transformation betwen H and $\widetilde{H}$, we can set $\Pi \to \Pi - M$ and $M \to 0$ to get a transformation where $M = 0$. Thus there exists a transformation iff there exist such cohomologies between the data. Such a transformation can be lifted to an equivalence by taking the inverse transformation to have the same data with a negative sign, and by taking $E = F = 0$.

1.8 Summary of results in $\beta = \mathrm{id}$ case

We can summarize all of the above in the following main results. Recall that C is a skeletal permutative Picard category with trivial braiding and B is an abelian group.

Theorem 1.24. *Any $H \in \mathrm{SymMonBiFun}(C, \Sigma^2 B)$ is equivalent to a symmetric monoidal bihomomorphism $\widetilde{H}$ in nice form, meaning one where*

- $\chi_{(f,a),(g,b)} = \chi_{(f,1),(g,1)}$
- $\gamma_a = \delta_a = 0$
- $\psi^a = 0$
- $\phi_{(f,a),(g,a)} = -\chi_{(f,1),(g,1)}$

Theorem 1.25. *A symmetric monoidal bihomomorphism H in nice form is determined uniquely by*

- *A normalized symmetric 2-cocycle χ with domain* $\mathrm{Aut}(1)$*, subject to the constraint that $\chi_{f,g} + \chi_{h,k} - \chi_{h\circ f,k\circ g} = \chi_{f,h} + \chi_{g,k} - \chi_{f\circ g,h\circ k}$.*
- *A normalized symmetric 3-cocycle (ω, u) with domain* $\mathrm{ob}\, C$*.*

Theorem 1.26. *Given two symmetric monoidal bihomomorphisms H and $\widetilde{H}$ in nice form, represented as in Theorem 1.25 by $(\chi, (\omega, u))$ and $(\widetilde{\chi}, (\widetilde{\omega}, \widetilde{u}))$ respectively, the following are equivalent:*

- *There exists a transformation $H \to \widetilde{H}$.*
- *H is equivalent to $\widetilde{H}$.*
- *χ is cohomologous to $\widetilde{\chi}$ and (ω, u) is cohomologous to $(\widetilde{\omega}, \widetilde{u})$.*

Note the implied sets of structures in the above theorem are not the same: transformations are not necessarily in one-to-one correspondence with cohomology pairs.

1.9 Obstructions when $\beta \neq \mathrm{id}$

We take a moment to consider why our previous result does not obviously work when $\beta \neq \mathrm{id}$. Given an arbitrary symmetric monoidal bihomomorphism H, when attempting to construct a nice symmetric monoidal bihomomorphism $\widetilde{H}$ that it is equivalent to, the following four constraints will not automatically be satisfied with the choices we made above:

- **F.10** $\widetilde{\omega}_{a,b,c} - \widetilde{\omega}_{b,a,c} + \widetilde{\omega}_{b,c,a} = \widetilde{u}_{b\otimes c,a} - \widetilde{u}_{b,a} - \widetilde{u}_{c,a} - \widetilde{\chi}_{\mathrm{id}_b,\beta_{a,c}} - \widetilde{\chi}_{\beta_{a,b},\mathrm{id}_c}$
- **F.11** $-\widetilde{\omega}_{a,b,c} + \widetilde{\omega}_{a,c,b} - \widetilde{\omega}_{c,a,b} = \widetilde{u}_{c,a\otimes b} - \widetilde{u}_{c,a} - \widetilde{u}_{c,b} - \widetilde{\chi}_{\beta_{a,c},\mathrm{id}_b} - \widetilde{\chi}_{\mathrm{id}_a,\beta_{b,c}}$
- **F.12** $\widetilde{\phi}_{\beta_{b,a},\beta_{a,b}} = \widetilde{\psi}^{a\otimes b} + \widetilde{u}_{b,a} + \widetilde{u}_{a,b}$
- **T.5** $\Pi_{a,b} - \Pi_{b,a} + \theta_{\beta_{a,b}} = \widetilde{u}_{b,a} - u_{b,a}$

We could hope to modify terms to make the above hold. **F.12** and **T.5** can be fixed by adding a term $-\chi_{(\beta_{b,a},1),(1,a\otimes b)}$ to $\widetilde{u}_{a,b}$, however it's unclear how one would then make **F.10** and **F.11** hold.

If we have two nice symmetric monoidal bihomomorphisms, the above characterization of their transformations also does not work; firstly, u is no longer obviously normalized due to the form of **F.12**. Secondly and more importantly, **T.5** is no longer the form for a cohomology due to the additional θ term.

1.10 Case $C = A$, an abelian group

If C is the discrete Picard category corresponding to an abelian group A, we can apply Theorem 1.25 since $\beta = \mathrm{id}$. In this case, since all 1-cells are identities, $\chi = 0$, so our nice symmetric monoidal bihomomorphisms are determined completely by a normalized symmetric 3-cocycle, (ω, u). Theorem 1.26 then says that two such symmetric monoidal bihomomorphisms are equivalent iff those symmetric 3-cocycles are cohomologous. Thus we've shown that we can identify $\mathrm{SymMonBiFun}(A, \Sigma^2 B)$ modulo equivalence with $H^3_{\mathrm{sym}}(A; B)$. This result was known previously from considerations on the structure of Picard 1-categories (see for instance [JS93, Section 3]; further references can be found in [JO12]).

Chapter 2

Computads and pattern matching

2.1 Data and axiom families

As outlined in [SP09], we can think of the data and axioms for a symmetric monoidal bihomomorphism, bitransformation, or modification as belonging to certain families related to familiar polyhedra. As an example, let $H : C \to D$ be a symmetric monoidal bifunctor between strict symmetric monoidal 2-categories. Part of the data for H is a bitransformation

$$\chi : H(a) \otimes H(b) \to H(a \otimes b) \tag{2.1}$$

along with a selected adjoint equivalence with adjoint $\chi^\bullet$. There is also an invertible modification

$$\begin{tikzcd}
 & H(a) \otimes H(b \otimes c) \arrow[dr, "\chi_{a,b\otimes c}"] & \\
H(a) \otimes H(b) \otimes H(c) \arrow[ur, "H(a)\otimes\chi_{b,c}"] \arrow[dr, "\chi_{a,b}\otimes H(c)"'] & \Downarrow\omega_{a,b,c} & H(a \otimes b \otimes c) \\
 & H(a \otimes b) \otimes H(c) \arrow[ur, "\chi_{a\otimes b,c}"'] &
\end{tikzcd} \qquad (2.2)$$

and an axiom that the 2-cell for the pasting diagram

$$\begin{tikzcd}
 & H(a)\otimes H(b)\otimes H(c\otimes d) \arrow[rr, "H(a)\otimes\chi_{b,c\otimes d}"] \arrow[d, phantom, "\Downarrow\ H(a)\otimes\omega_{b,c,d}"] & & H(a)\otimes H(b\otimes c\otimes d) \arrow[dr, "\chi_{a,b\otimes c\otimes d}"] & \\
H(a)\otimes H(b)\otimes H(c)\otimes H(d) \arrow[ur, "H(a)\otimes H(b)\otimes\chi_{c,d}"] \arrow[rr, "H(a)\otimes\chi_{b,c}\otimes H(d)"] \arrow[dr, "\chi_{a,b}\otimes H(c)\otimes H(d)"'] & & H(a)\otimes H(b\otimes c)\otimes H(d) \arrow[ur, "H(a)\otimes\chi_{b\otimes c,d}"] \arrow[dr, "\chi_{a,b\otimes c}\otimes H(d)"'] \arrow[rr, phantom, "\Downarrow\ \omega_{a,b\otimes c,d}"] & & H(a\otimes b\otimes c\otimes d) \\
 & H(a\otimes b)\otimes H(c)\otimes H(d) \arrow[rr, "\chi_{a\otimes b,c}\otimes H(d)"] \arrow[u, phantom, "\Downarrow\ \omega_{a,b,c}"] & & H(a\otimes b\otimes c)\otimes H(d) \arrow[ur, "\chi_{a\otimes b\otimes c,d}"'] &
\end{tikzcd} \tag{2.3}$$

is equal to that for

$$\begin{tikzcd}
 & H(a)\otimes H(b)\otimes H(c\otimes d) \arrow[rr, "H(a)\otimes\chi_{b,c\otimes d}"] \arrow[dr, "\chi_{a,b}\otimes H(c\otimes d)"] & & H(a)\otimes H(b\otimes c\otimes d) \arrow[dr, "\chi_{a,b\otimes c\otimes d}"] \arrow[d, phantom, "\Downarrow\ \omega_{a,b,c\otimes d}"] & \\
H(a)\otimes H(b)\otimes H(c)\otimes H(d) \arrow[ur, "H(a)\otimes H(b)\otimes\chi_{c,d}"] \arrow[dr, "\chi_{a,b}\otimes H(c)\otimes H(d)"'] & = & H(a\otimes b)\otimes H(c\otimes d) \arrow[rr, "\chi_{a\otimes b,c\otimes d}"] & & H(a\otimes b\otimes c\otimes d) \\
 & H(a\otimes b)\otimes H(c)\otimes H(d) \arrow[ur, "H(a\otimes b)\otimes\chi_{c,d}"] \arrow[rr, "\chi_{a\otimes b,c}\otimes H(d)"] & & H(a\otimes b\otimes c)\otimes H(d) \arrow[ur, "\chi_{a\otimes b\otimes c,d}"'] \arrow[u, phantom, "\Downarrow\ \omega_{a\otimes b,c,d}"] &
\end{tikzcd} \tag{2.4}$$

Each of these clearly relates to an increasing number of objects $a, b, c, \ldots \in C$ and how they interact with bifunctor and tensor applications. If C, D are instead abelian groups and H a group homomorphism, then the analogue of Eq. (2.1) is an equality, as are the 1-boundaries of the other two diagrams. We can think of this as a 0-dimensional analogue of a symmetric monoidal 2-category.

If C, D are symmetric monoidal 1-categories and H a (lax) symmetric monoidal functor, then the analogue of χ is now a natural transformation (instead of a bitransformation), and ω is instead an axiom relating its two boundaries. Thus, we can observe the pattern that at each increase in dimension, the axiom at the previous dimension is replaced with a new piece of data, and a new axiom is added one dimension higher.

In diagram Eq. (2.2), notice that the upper and lower boundaries represent both possible orderings of combining tensors underneath applications of H starting from $H(a) \otimes H(b) \otimes H(c)$ to $H(a \otimes b \otimes c)$, and every possible combination appears as an object along the boundary. This is no longer true for Eq. (2.3) and Eq. (2.4); for instance $H(a) \otimes H(b \otimes c) \otimes H(d)$ does not appear as an object in the pasting diagram Eq. (2.4), but does in Eq. (2.3).

One way to view this situation is that the corresponding directed graph of 1-cells is topologically a circle in the first case, but not in the second. However, in the second case, the directed polytope coming from all possible fillings of that graph by instances of ω and functoriality equivalences (potentially tensored with identities) forms a 2-sphere. Thus any two distinct paths from $H(a) \otimes H(b) \otimes H(c) \otimes H(d) \to H(a \otimes b \otimes c \otimes d)$ witnessed in Eq. (2.3) or Eq. (2.4) will yield a decomposition of the polytope into two pieces. Since ω is invertible, there is a 2-cell associated to the corresponding pasting diagram for each piece. We could have equivalently expressed the axiom by requiring any such choice of 2-cells to be equal.

Similar patterns are repeated for the other data and axioms for a strict symmetric monoidal 2-category, as well as for symmetric monoidal bihomomorphisms, bitransformations, and modifications between such.

2.2 Boundary filling

If we happened to forget the source and target of $\omega_{a,b,c}$ as exhibited in Eq. (2.2), an obvious method of reconstruction might occur to us: find all paths via 1-cells from $H(a) \otimes H(b) \otimes H(c)$ which involve only tensor products, identities, and χ. Since there are only two such paths, one will be the source and the other the target. Since ω is invertible, if we were to swap the source and target we would have an equivalent definition of symmetric monoidal bicategory; just switch ω for ω^{-1} to change between them.

We could apply a similar method to the axiom Eq. (2.3) = Eq. (2.4); instead of finding 1-cell paths between given 0-cells, we need to find 2-cell paths between given 1-cells. There is one additional complication, in that there's no obvious canonical choice for which 1-cells to take as the source and target. There is only one choice between which there are two distinct 2-cell paths without using ω^{-1}, so this is the natural one. Similarly for Eq. (2.2), there is only one choice of 0-cells from the diagram between which their exists two distinct 1-cell paths without the use of $\chi^{\bullet}$.

In the rest of this chapter, we put this process on firm footing. We'll make use of the language of computads to describe what we mean by finding all paths involving certain families of cells. Then we'll prove results about decompositions in certain low dimensional computads that allow us to do so algorithmically. Specifically, we will work with the following class of computads:

Definition 2.1. A **strict monoidal n-computad** is an $(n+1)$-computad with a single 0-cell $*$.

All 1-cells (in the usual computad sense) in a strict monoidal n-computad will compose since they must share source/target – this composition represents a monoidal/tensor product. In the remainder, we renumber to call this unique 0-cell a (-1)-cell in order to preserve intuitions about dimension; that is, what we call a k-cell in a strict monoidal n-computad is actually a $k+1$-cell in its representing $(n+1)-$computad.

In a strict monoidal n-comptuad, composition along the unique (-1)-cell boundary will be denoted by $\otimes$, composition along a 0-cell boundary will be denoted by $\star$, and composition along a 1-cell boundary will be denoted by $\circ$. Note that we'll be using the **geometric order** of composition throughout; if f and g are 1-cells, then $f \star g$ is the 1-cell coming from first following f, then g. We'll denote the source operator on an n-cell by s and the target operator by t. If $k < n$, we'll denote the k-dimensional source (resp. target) by $\mathrm{s}^k(x)$ (resp. $\mathrm{t}^k(x)$). For example, if x is a 2-cell, $\mathrm{s}^0(x) = \mathrm{s}(\mathrm{s}(x)) = \mathrm{s}(\mathrm{t}(x))$.

2.3 Primitive families

In our initial example, ω was not just an individual primitive; it was a family of primitives $\omega_{a,b,c}$ indexed by triplets of 0-cells and obeying a modification axiom. Consider the 1-cell boundary:

$$\begin{tikzcd}
H(a) \otimes H(b) \otimes H(c) \otimes H(1) \otimes H(d) \arrow[r, "H(a) \otimes H(b) \otimes \chi_{c,1} \otimes H(d)"] & H(a) \otimes H(b) \otimes H(c \otimes 1) \otimes H(d) \arrow[d, "\chi_{a,b} \otimes H(c \otimes 1) \otimes H(d)"] \\
& H(a \otimes b) \otimes H(c \otimes 1) \otimes H(d) \arrow[d, "\chi_{a \otimes b, c \otimes 1} \otimes H(d)"] \\
& H(a \otimes b \otimes c \otimes 1) \otimes H(d)
\end{tikzcd}$$

By observation, we can notice that this is the source of a 2-cell involving omega:

$$\begin{tikzcd}
H(a) \otimes H(b) \otimes H(c) \otimes H(1) \otimes H(d) \arrow[r, "H(a) \otimes H(b) \otimes \chi_{c,1} \otimes H(d)"] \arrow[d, "\chi_{a,b} \otimes H(c) \otimes H(1) \otimes H(d)"'] \arrow[dr, phantom, "="] & H(a) \otimes H(b) \otimes H(c \otimes 1) \otimes H(d) \arrow[d, "\chi_{a,b} \otimes H(c \otimes 1) \otimes H(d)"] \\
H(a \otimes b) \otimes H(c) \otimes H(1) \otimes H(d) \arrow[r, "H(a \otimes b) \otimes \chi_{c,1} \otimes H(d)"] \arrow[d, "\chi_{a \otimes b, c} \otimes H(1) \otimes H(d)"'] \arrow[dr, phantom, "\swarrow \omega_{a \otimes b, c, 1} \otimes H(d)"] & H(a \otimes b) \otimes H(c \otimes 1) \otimes H(d) \arrow[d, "\chi_{a \otimes b, c \otimes 1} \otimes H(d)"] \\
H(a \otimes b \otimes c) \otimes H(1) \otimes H(d) \arrow[r, "\chi_{a \otimes b \otimes c, 1} \otimes H(d)"] & H(a \otimes b \otimes c \otimes 1) \otimes H(d)
\end{tikzcd}$$

This illustrates several issues that arise when determining whether a given 1-cell is the source of a given instance of ω:

(i) The parameters to ω can be any 0-cells; we must determine which are viable.

(ii) We may need to re-express part of the boundary as an equivalent 1-cell in order to recognize it as the source of an instance of ω.

(iii) The source of such an ω may only involve a subset of the 1-cells in a diagram exhibiting the boundary.

(iv) The source of such an ω may only involve a subset of the tensored components of the 1-cell exhibiting the boundary.

Definition 2.2. Let C be a strict monoidal n-computad; for each k let C_k be its set of k-cells and let C_k^{prim} be its set of k-primitives. Define a **primitive family of dimension** $m \leq n$ inductively as follows: for each $0 \leq i < m$, choose an arity A_i, and for each $0 \leq j < A_i$, let x_j^i be an indeterminate symbol whose dimension we will call i. Then a primitive family α is an injective map

$$C_0^{A_0} \times \cdots \times C_{m-1}^{A_{m-1}} \to C_m^{\text{prim}}$$

along with two well-typed abstract syntax trees S and T of dimension $m-1$ in the following symbol set:

- Primitive families of dimension less than m, considered as operations of the appropriate arity.
- For all $0 \leq k \leq \ell < m$, k-dimensional composition of ℓ-dimensional symbols.
- For all k, identities as operations from dimension k symbols to dimension $k+1$ symbols.
- C_k for all k.

- Indeterminates x^i_j.

- Sources and targets of nodes.

such that given a tuple $\mathbf{x}$ of the appropriate arity consisting of cells in C, the source of $\alpha(\mathbf{x})$ is the $m-1$-cell coming from evaluating the AST S under substitution for the x^i_j's, and similarly for the target and T. Such a primitive family is **source-solvable** if each indeterminate occurs at least once in the source AST while not underneath a source or target node, and all source/target nodes only have source/target nodes and indeterminates as subnodes. **Target-solvable** is defined similarly, and if the family is both source and target solvable we'll simply call it **solvable**.

The trick in the above definition is to select a particular *syntactic form* for the source and target of each primitive in the family. As an example, we will think of ω not just as a single primitive $\omega_{a,b,c}$ for each choice of a, b, and c – we will also choose a particular strategy for constructing its source and target. For example, for its source we can take, $(H(a) \otimes \chi_{b,c}) \star (\chi_{a,b\otimes c})$ with the obvious order of operations.

2.4 Pattern matching

Issues (iii) and (iv) above are similar; given a boundary cell b, we may be able to express b as a composite of other cells, including some given cell c. If c is the source of an instance of a primitive family, then b is the source of an appropriately whiskered/tensored instance of it.

In general, for dimension k cells we will have k types of composition. If we could determine all ways in which a given k-cell can be expressed as a non-trivial composition, then we could reduce to the problem of recognizing when a cell is exactly the source of some instance.

Source-solvable primitive families give us an approach to doing this, which also addresses issues (i) and (ii) – to determine which members of a source-solvable primitive family a given k-cell is the source of, we can pattern match against the source AST of the family. This again relies on sub-algorithms for finding all decompositions of a k-cell into ℓ-dimensional compositions for $0 \leq \ell \leq k$.

Thus, we can break our algorithm into two parts. We need to be able to find lists of decompositions as mentioned above, and we need to be able to determine when a cell is exactly the source of an instance of a primitive family. The later part of this chapter is devoted to proving the required decomposition results; see Theorem 2.44, Theorem 2.65, and Theorem 2.73. Given these, we can describe the matching algorithm as follows:

Algorithm 2.3 (Matching cells to primitive families)**.** Start with an n-cell in a non-degenerate strict monoidal n-computad, and a source abstract syntax tree of a source-solvable primitive family of dimension n as in Definition 2.2. We will construct an exhaustive list of mappings from indeterminates to cells, such that evaluating the AST by replacing indeterminates with their values gives the starting n-cell.

Let a **parameter map** be a partial mapping from indeterminates to cells of the appropriate dimension, along with a partial mapping from iterated source/targets of indeterminates (as symbols) to cells of the appropriate dimension, such that the choices of source/-target cells are coherent with themselves and choices of indeterminate cells. Our algorithm will initially return a list of parameter maps; we will then discard the source/target portion, and return any parameter maps which provide an assignment for each indeterminate in the AST.

Say that two parameter maps are compatible if their assignments are coherent with one another, and define their unification in the obvious way.

We start at the top node in our AST, comparing the given n-cell to this node's symbol. Let α be our cell, and S be our symbol. Proceed as follows:

- If S is a primitive family symbol, and α is not a primitive of that family, then return no matches. If it is a primitive of that family, recurse on each of the subnodes of that family using the parameters of α (which are uniquely determined by injectivity). Then return all unifications of compatible choices from each subresult.

- If S is an ℓ-dimensional composition symbol, take all ℓ-dimensional decompositions of α. For each decomposition pair, recurse against each subnode. Return the union across all decompositions of unifications of compatible choices from each subresult in a given pair.
- If S is an identity symbol and α is not an identity cell, return no matches. Otherwise, return the result of recursing on the subnode of S against the source of α.
- If S is a fixed primitive symbol and α is not that primitive, return no matches. Otherwise, return a single empty parameter map.
- If S is an indeterminate symbol x^i_j, return a single parameter map matching x^i_j to α.
- If S is an iterated source/target of some x^i_j, return a single parameter map matching that iterated source/target symbol string to α.

2.5 Atoms, molecules, and equivalence checking

The cells of a strict monoidal n-computad (or computad in general) are in some sense freely generated from its primitives under composition operations and identities. However, the categorical structure still imposes constraints on the cells – in general, they can have multiple such representations.

Example 2.4. If $\mathbf{f}$ is a 1-primitive with source a and target b, and $\mathbf{m}$ is a 2-primitive with 0-source c and 0-target d, then $(a \otimes \mathbf{m}) \star (f \otimes d) = (f \otimes c) \star (b \otimes \mathbf{m})$.

In the remainder of this section, we will follow [Mak05] where the author defines "syntactic representatives" for n-cells, called n-molecules. Each n-molecule will represent one way of constructing an n-cell, and every n-cell will have at least one n-molecule representative. Furthermore, the paper concludes with a computable algorithm for determining when two n-molecules represent the same n-cell. We will first summarize the results of [Mak05], rephrased and restricted to the case of a strict monoidal 2-computad. Then, we will diverge from [Mak05] to give a more detailed account of equivalence checking of n-molecules in (non-degenerate, cf. Definition 2.23) strict monoidal 2-computads. We will use this account to prove our main theorems – recognition principles for when certain molecules can be decomposed as products; these are given in Theorem 2.44, Theorem 2.65, and Theorem 2.73.

When necessary we will use $|f|$ to indicate the underlying n-cell of an atom or molecule f (defined below). We will say that two atoms or molecules are **equivalent** if they represent equal n-cells; in symbols, $x \cong y$ iff $|x| = |y|$. We will reuse the symbols $\star$, $\otimes$, and $\circ$ in the notation for atoms and molecules, as well as in the definition of operations on them – in practice, the usage will be clear from context.

Definition 2.5.

- Define a 0-**atom** to be a single 0-primitive.
- Define a 0-**molecule** to be either:
  - A finite non-empty list of 0-atoms/0-primitives, denoted $\mathbf{a_0} \otimes \cdots \otimes \mathbf{a_k}$.
  - Or an "empty" 0-molecule denoted $\emptyset$ (representing the identity on the unique (-1)-cell).
- Define a 1-**atom** to be a pair of 0-molecules a, b and a 1-primitive $\mathbf{f}$, denoted $a \otimes \mathbf{f} \otimes b$.
- Define a 1-**molecule** to be either:
  - A finite non-empty list of 1-atoms, denoted $f_0 \star \cdots \star f_k$, where for each adjacent pair of atoms the target of the 1-cell represented by the leftmost atom is equal to the source of the 1-cell represented by the rightmost.

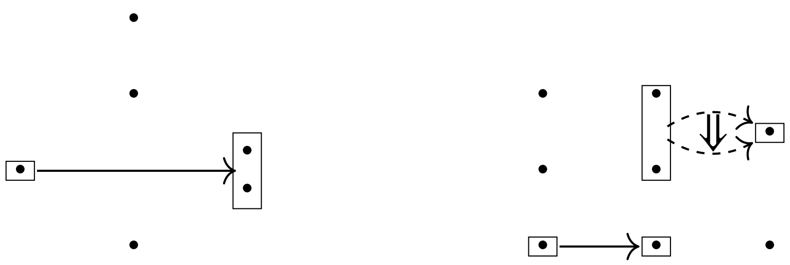

Figure 2.1: Illustrations of a 1-atom (left) and 2-atom (right). Empty vertical space respresents tensor, read from top to bottom. Solid arrows and dots are primitives, whereas the dashed arrows in the 2-atom are themselves 1-molecules. Rectangles are drawn around 0-sources and 0-targets. Labels are omitted. The 1-atom on the left is $a_0 \otimes \mathbf{p_0} \otimes b_0$ where a_0 is a 0-molecule of length 2, and b_0 is a 0-molecule of length 1. The 2-atom on the right is $l_1 \star (a_1 \otimes \mathbf{p_1} \otimes b_1) \star r_1$ where l_1 is a 1-molecule of length 1, r_1 is an identity 1-molecule, $a_1 = \emptyset$, and b_1 is a 0-molecule of length 1.

  - Or a single 0-molecule a, denoted 1_a

- Define a 2-**atom** to be a pair of 0-molecules a, b, a 2-primitive $\mathbf{m}$, and a pair of 1-cells l, r denoted $l \star (a \otimes \mathbf{n} \otimes b) \star r$ such that $\mathrm{t}(|l|) = |a| \otimes \mathrm{s}^0(|\mathbf{m}|) \otimes |b|$ and $\mathrm{s}(|r|) = |a| \otimes \mathrm{t}^0(|\mathbf{m}|) \otimes |b|$.

- Define a **2-molecule** to be either:

  - A finite non-empty list of 2-atoms, denoted $f_0 \circ \cdots \circ f_k$, where for each adjacent pair of atoms the codomain of the 2-cell represented by the leftmost atom is equivalent to the domain of the 2-cell represented by the rightmost.

  - Or a single 1-molecule f, denoted 1_f.

Next we define versions of each categorical composition which act on n-molecules instead of n-cells, for $n = 0, 1, 2$. Each of these operations will commute with the underlying cell function. We could make different choices with respect to k-composition of

n-molecules for $k < n$; the definitions below are chosen with the most obvious "front-to-back, left-to-right" orientation.

Definition 2.6. Define compositions of n-molecules for $n = 0, 1, 2$ as follows:

- For 0-molecules:
  - If $f = \mathbf{a_1} \otimes \cdots \otimes \mathbf{a_k}$, $g = \mathbf{b_1} \otimes \cdots \otimes \mathbf{b_m}$, then $f \otimes g := \mathbf{a_1} \otimes \cdots \otimes \mathbf{a_k} \otimes \mathbf{b_1} \otimes \cdots \otimes \mathbf{b_m}$
  - If $f = \mathbf{a_1} \otimes \cdots \otimes \mathbf{a_k}$, $g = \emptyset$, then $f \otimes g := f$, $g \otimes f := f$.
  - If $f = \emptyset$, $g = \emptyset$, $f \otimes g := \emptyset$.
- For a 1-atom $f = a \otimes \mathbf{f} \otimes b$ and a 0-molecule c, define 1-atoms $f \otimes c := a \otimes \mathbf{f} \otimes (b \otimes c)$, and $c \otimes f := (c \otimes a) \otimes \mathbf{f} \otimes b$
- For 1-molecules:
  - If $f = f_1 \star \cdots \star f_k$, $g = g_1 \star \cdots \star g_m$, then:
    - $f \star g := f_1 \star \cdots \star f_k \star g_1 \star \cdots \star g_m$
    - $f \otimes g := (f_1 \otimes \mathrm{s}(g)) \star \cdots \star (f_k \otimes \mathrm{s}(g)) \star (\mathrm{t}(f) \otimes g_1) \star \cdots \star (\mathrm{t}(f) \otimes g_m)$
  - If $f = f_1 \star \cdots \star f_k$, $g = 1_a$, then:
    - $f \star g := f$, $g \star f := f$
    - $f \otimes g := (f_1 \otimes a) \star \cdots \star (f_k \otimes a)$, $g \otimes f := (a \otimes f_1) \star \cdots \star (f_k \otimes a)$
  - If $f = 1_a$, $g = 1_b$, then:
    - $f \star g := f$.
    - $f \otimes g := 1_{a \otimes b}$.
- For a 2-atom $f = l \star (a \otimes \mathbf{f} \otimes b) \star r$ and a 0-molecule c, define 2-atoms $f \otimes c := (l \otimes 1_c) \otimes (a \otimes \mathbf{f} \otimes (b \otimes c)) \otimes (r \otimes 1_c)$, and $c \otimes f := (1_c \otimes l) \otimes ((c \otimes a) \otimes \mathbf{f} \otimes b) \otimes (1_c \otimes r)$.
- For a 2-atom $f = l \star (a \otimes \mathbf{f} \otimes b) \star r$ and a 1-molecule g, define 2-atoms:

  - $f \star g := l \star (a \otimes \mathbf{f} \otimes b) \star (r \star g)$ and $g \star f := (g \star l) \star (a \otimes \mathbf{f} \otimes b) \star r$.
  - $f \otimes g := (f \otimes \mathrm{s}(g)) \star (\mathrm{t}^0(f) \otimes g)$ and $g \otimes f := (g \otimes \mathrm{s}^0(f)) \star (\mathrm{t}(g) \otimes f)$.

- For 2-molecules:
  - If $f = f_1 \circ \cdots \circ f_k$, $g = g_1 \circ \cdots \circ g_m$, then:
    - $f \circ g := f_1 \circ \cdots \circ f_k \circ g_1 \circ \cdots \circ g_m$.
    - $f \star g := (f_1 \star \mathrm{s}(g)) \circ \cdots \circ (f_k \star \mathrm{s}(g)) \circ (\mathrm{t}(f) \star g_1) \circ \cdots \circ (\mathrm{t}(f) \star g_m)$.
    - $f \otimes g := (f_1 \otimes \mathrm{s}(g)) \circ \cdots \circ (f_k \otimes \mathrm{s}(g)) \circ (\mathrm{t}(f) \otimes g_1) \circ \cdots \circ (\mathrm{t}(f) \otimes g_m)$
  - If $f = f_1 \circ \cdots \circ f_k$, $g = 1_h$, then:
    - $f \circ g := f$ and $g \circ f := f$
    - $f \star g := (f_1 \star h) \circ \cdots \circ (f_k \star h)$ and $g \star f := (h \star f_1) \circ \cdots \circ (h \star f_k)$
    - $f \otimes g := (f_1 \otimes h) \circ \cdots \circ (f_k \otimes h)$ and $g \otimes f := (h \otimes f_1) \circ \cdots \circ (h \otimes f_k)$
  - If $f = 1_h$ and $g = 1_\ell$ then:
    - $f \circ g := f$
    - $f \star g := 1_{h \star \ell}$
    - $f \otimes g := 1_{h \otimes \ell}$

Definition 2.7. A 0-molecule x is said to be a **prefix** of a 1-atom $a \otimes \mathbf{f} \otimes b$ if it is a prefix of a as a list, or it is an identity with the appropriate domain. A 1-molecule x is said to be a **prefix** of a 2-atom $l \star (a \otimes \mathbf{f} \otimes b) \star r$ if it is a prefix of l as a list, or it is an identity with the appropriate domain. Suffixes are defined similarly

Theorem 2.8 (Makkai). *Every n-cell can be represented by an n-molecule*

Proof outline. Proceed inductively. Since n-cells are freely generated, every n-cell can be described via compositions applied to n-primitives and identities of $(n-1)$-cells. We can represent each of these by an n-molecule, and then apply n-molecule versions of each composition operation (as in Definition 2.6).

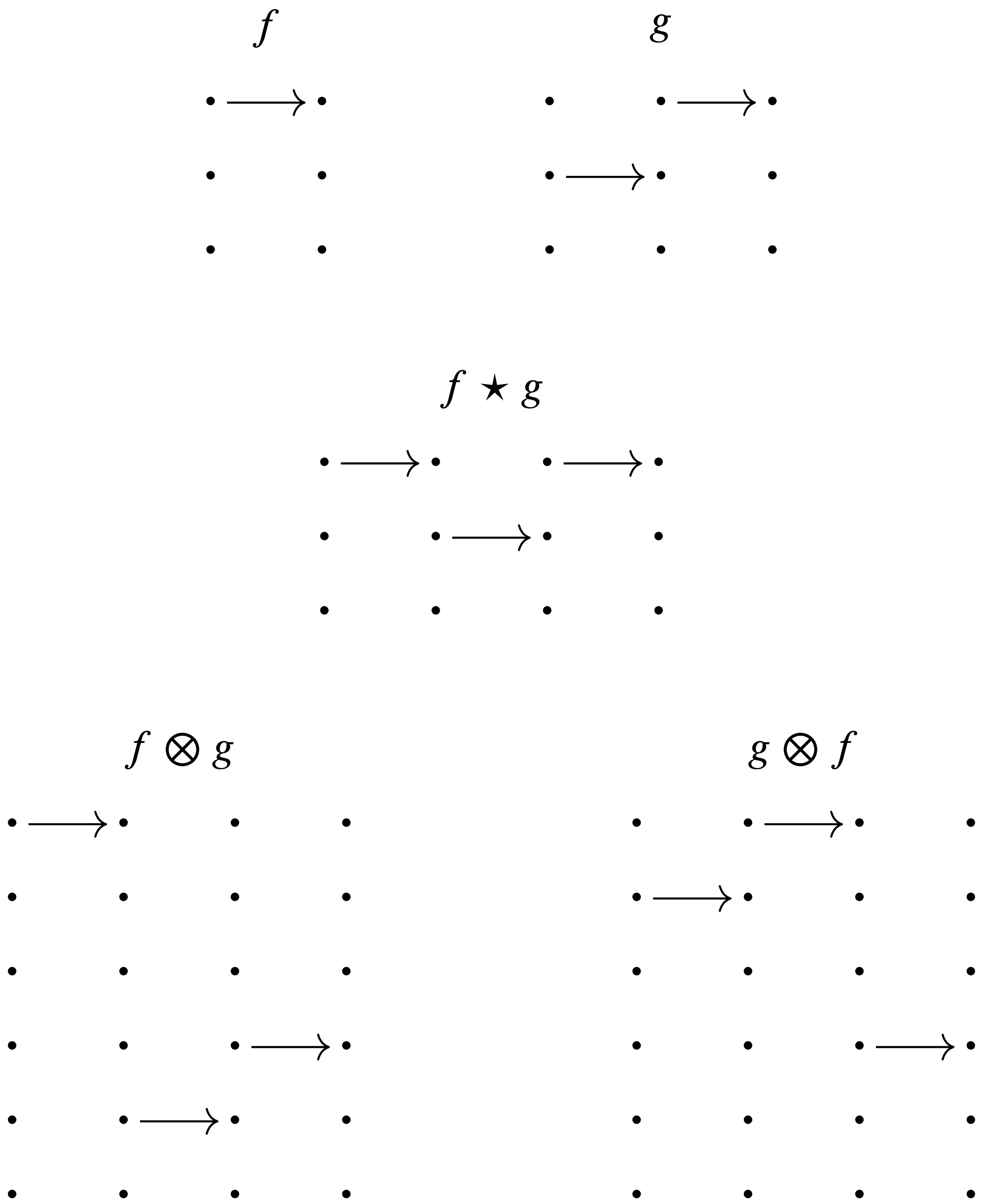

Figure 2.2: Illustrations without labels of 1-molecule compositions. Vertical separation is tensor, solid arrows are 1-primitives, bullets are 0-primitives.

Definition 2.9. The **index** of an n-atom in an n-molecule is its position in the n-molecule list, starting from 0.

Definition 2.10. The **length** $\mathrm{length}(f)$ of an n-molecule f is the number of atoms in f if it is a non-identity n-molecule and 0 if it is an identity n-molecule.

Definition 2.11. A **submolecule** of an n-molecule f is either:

- A contiguous subset of atoms in the molecule, such that they themselves form a non-identity molecule, or
- An identity molecule $1_{\mathrm{s}(f_i)}$ or $1_{\mathrm{t}(f_i)}$ for some $0 \leq i < \mathrm{length}(f)$.

Definition 2.12. Let x, y and z, w be adjacent pairs of n-atoms; that is, $\mathrm{t}(x) = \mathrm{s}(y)$ and $\mathrm{t}(z) = \mathrm{s}(w)$. Say that $L(x, y, z, w)$ holds if there exist n-atoms α and β such that $x \cong \alpha \star \mathrm{s}(\beta)$, $y \cong \mathrm{t}(\alpha) \star \beta$, $z \cong \mathrm{s}(\alpha) \star \beta$, and $w \cong \alpha \star \mathrm{t}(\beta)$.

Definition 2.13. Let $f = f_0 \star \cdots \star f_k$ and $g = g_0 \star \cdots \star g_\ell$ be two non-identity 1-molecules. We say that $E(f, g)$ holds if $k = \ell$ and there exists i such that either $L(f_i, f_{i+1}, g_i, g_{i+1})$ or $L(g_i, g_{i+1}, f_i, f_{i+1})$ holds. Define $\sim$ to be the equivalence relation generated by E.

Definition 2.14. Two n-molecules are called **atom equivalent** if they are the same length and their corresponding atoms are all equivalent.

Definition 2.15. Given a 2-atom $x = l \star (a \otimes \mathbf{f} \otimes b) \star r$, let $\bar{\mathbf{f}}$ be a newly adjoined 1-primitive with $\mathrm{s}\left(\bar{\mathbf{f}}\right) = \mathrm{s}^0(\mathbf{f})$ and $\mathrm{t}\left(\bar{\mathbf{f}}\right) = \mathrm{t}^0(\mathbf{f})$. The **collapse** of x (denoted $\overline{x}$) is then the 1-molecule $l \star (a \otimes \bar{\mathbf{f}} \otimes b) \star r$.

Theorem 2.16 (Makkai)**.**

- *Two 0-molecules are equivalent iff they are equal as lists of primitives.*
- *Two identity 1-molecules 1_a and 1_b are equivalent iff $a = b$. No identity 1-molecule is equivalent to a non-identity 1-molecule.*

- *Two 1-atoms are equivalent iff they are equal.*

- *Two non-identity 1-molecules f and g are equivalent iff $f \sim g$.*

- *Two identity 2-molecules 1_f and 1_g are equivalent iff $f \cong g$. No identity 2-molecule is equivalent to a non-identity 2-molecule.*

- *Two 2-atoms x_1 and x_2 are equivalent iff $\overline{x_1} \cong \overline{x_2}$ as 1-molecules.*

- *Two non-identity 2-molecules f and g are equivalent iff there exist atom-equivalent 2-molecules f' and g' such that $f' \sim g'$.*

2.6 Tags and spans

Definition 2.17. A **tagged n-molecule** is an n-molecule of length k along with a list of length k composed of distinct elements from some indexing set I. We call this list the **tag list**, and individual elements of this list are **tags**. For tags i, j, say that $i \prec j$ if i precedes j in the corresponding tag list.

We can extend composition to tagged n-molecules by taking a disjoint union of their indexing sets if necessary. We can upgrade any n-molecule to a tagged n-molecule by tagging it by indices (if an indexing set is not otherwise implied).

Definition 2.18. A **span** (i, j) for an n-molecule f is a pair of integers $i \leq j$ such that $0 \leq i, j \leq \text{length}(f)$. Note that a span for an n-molecule selects a submolecule, potentially an identity submolecule if $i = j$.

Definition 2.19. Define the **spanned source of a 1-atom** $g = a \otimes \mathbf{f} \otimes b$ as the pair of the 0-molecule $\mathrm{s}(g)$ and the span corresponding to $\mathrm{s}(\mathbf{f})$. That is, if $\text{length}(a) = i$ and $\text{length}(\mathrm{s}(\mathbf{f})) = j$, it is the pair $(\mathrm{s}(g), (i, i + j))$. Define the **spanned source of a 2-atom** $g = l \star (a \otimes \mathbf{m} \otimes b) \star r$ as the 1-molecule $\mathrm{s}(g)$ along with the span corresponding to $a \otimes \mathrm{s}(\mathbf{m}) \otimes b$. Define the **spanned targets** similarly.

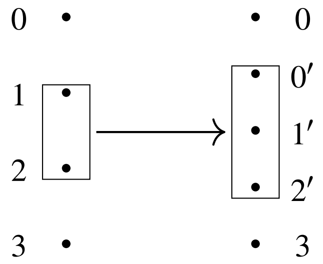

Figure 2.3: The spanned source and target of a 1-atom. Boxes indicate the domain and codomain of the primitive. The labels on the left are a tagging of the source by indices. The source span is $(1, 3)$. The labels on the right are a tag list for the target coming from substitution; note the choice of disjoint labels $0', 1', 2'$. The target span is $(1, 4)$.

Definition 2.20. Let f, g be n-molecules and (i, j) a span for f. If $\mathrm{s}(g) = \mathrm{s}\left(f_i\right)$ and $\mathrm{t}(g) = \mathrm{t}\left(f_{j-1}\right)$, we define the **substitution** of g for (i, j) to be the n-molecule coming from deleting the atoms with indices k s.t. $i \leq k < j$ and replacing them with the atoms of g. If f is tagged, we define a tag on the substitution by selecting new disjoint indices for the atoms of g.

Lemma 2.21. *The source and target of an n-atom ($n = 1, 2$) can be constructed from one another by substitution.*

Proof. This is obvious from the definitions. □

2.7 Non-degeneracy

The structure of 1-molecules turns out to be much easier to analyze in the case where no 1-primitive has an empty source or target. As an example, consider 0-molecules a, b, c; then $a \otimes b \cong a \otimes c$ if and only if $b \cong c$ since equivalence of 0-molecules is given by equality as lists of primitives. We would like to be able to use the same result for 1-molecules, but unfortunately it is untrue:

Example 2.22. Let $\mathbf{p}$ and $\mathbf{q}$ be distinct 1-primitives with $\mathrm{s}(\mathbf{q}) = a$, $\mathrm{s}(\mathbf{p}) = \mathrm{t}(\mathbf{p}) = \mathrm{t}(\mathbf{q}) = \emptyset$ for some 0-molecule $a \neq \emptyset$. Then $(a \otimes \mathbf{p} \otimes \emptyset) \star (\emptyset \otimes \mathbf{q} \otimes \emptyset) \cong (\emptyset \otimes \mathbf{q} \otimes \emptyset) \star (\emptyset \otimes \mathbf{p} \otimes \emptyset) \cong (\emptyset \otimes \mathbf{p} \otimes a) \star (\emptyset \otimes \mathbf{q} \otimes \emptyset)$, but $(a \otimes \mathbf{p} \otimes \emptyset) \ncong (\emptyset \otimes \mathbf{p} \otimes a)$.

The desired result is true if we restrict sources and targets; we prove this in Theorem 2.51. This motivates the following definition:

Definition 2.23. Call a 1-primitive **non-degenerate** if neither its source nor its target are $\emptyset$. Call a 1-atom **non-degenerate** if its primitive is non-degenerate. Call a 1-molecule **non-degenerate** if it is 1_a for some $a \neq \emptyset$, or if all of its atoms are non-degenerate. Call a strict monoidal n-computad **non-degenerate** if all of its 1-primitives are non-degenerate.

The non-degenerate case turns out to be sufficient for our desired application to pattern-matching in strict Picard 2-categories; thus throughout the remainder of this chapter, we will take non-degeneracy as a standing assumption:

Standing Assumption 2.24. All 1-primitives are non-degenerate.

2.8 Equivalence checking in dimension 1

Equivalence checking in dimension 0 is already as simple as possible (equality of lists of primitives), so we move on to equivalence checking in dimension 1.

Definition 2.25. We will say that two adjacent 1-atoms in a 1-molecule, denoted $f_i = a_i \otimes \mathbf{f_i} \otimes b_i$ for $i = 1, 2$, are **left transposable** if there exist 1-atoms $\alpha = a_\alpha \otimes \mathbf{f_1} \otimes b_\alpha$ and $\beta = a_\beta \otimes \mathbf{f_2} \otimes b_\beta$ such that $f_1 = \alpha \otimes \mathrm{s}(\beta)$ and $f_2 = \mathrm{t}(\alpha) \otimes \beta$. The **left transposition** is then defined to be the new pair of adjacent 1-atoms $g_1 = \mathrm{s}(\alpha) \otimes \beta, g_2 = \alpha \otimes \mathrm{t}(\beta)$. Say that the two 1-atoms are **right transposable** if there exist α and β such that $f_1 = \mathrm{s}(\alpha) \otimes \beta$ and $f_2 = \alpha \otimes \mathrm{t}(\beta)$. Define the **right transposition** to be the new pair of adjacent 1-atoms $g_1 = \alpha \otimes \mathrm{s}(\beta)\,, g_2 = \mathrm{t}(\alpha) \otimes \beta$. Say that two 1-atoms are **transposable** if they are either left or right transposable.

Lemma 2.26. *Two 1-atoms $f_i = a_i \otimes \mathbf{f_i} \otimes b_i$ for $i = 1, 2$ are left transposable iff there exists a 0-molecule x such that $a_2 = a_1 \otimes \mathrm{t}(\mathbf{f_1}) \otimes x$ and $b_1 = x \otimes \mathrm{s}(\mathbf{f_2}) \otimes b_2$. They are right transposable iff there exists a 0-molecule x such that $a_1 = a_2 \otimes \mathrm{s}(\mathbf{f_2}) \otimes x$ and $b_2 = x \otimes \mathrm{t}(\mathbf{f_1}) \otimes b_1$.*

Proof. We prove this only for left transposability; the right transposability case is analagous. By pattern matching in the definition, we see that the two 1-atoms being left transposable via a given α and β is equivalent to:

$$a_1 = a_\alpha$$

$$b_2 = b_\beta$$

$$a_2 = a_\alpha \otimes \mathrm{t}(\mathbf{f_1}) \otimes b_\alpha \otimes a_\beta$$

$$b_1 = b_\alpha \otimes a_\beta \otimes \mathrm{s}(\mathbf{f_2}) \otimes b_\beta$$

Thus, if they are transposable, we can take $x = b_\alpha \otimes a_\beta$.

If there exists such an x, choose

$$a_\alpha = a_1$$

$$b_\alpha = \emptyset$$

$$a_\beta = x$$

$$b_\beta = b_2$$

□

Lemma 2.27. *If f and g are transposable 1-atoms, they have a unique transposition.*

Proof. If f and g are left transposable, then we can see from the definition that their left transposition is determined by the selection of x in Lemma 2.26; different α, β yield the same transposition as long as the corresponding x is the same. If x_1 and x_2 are two such x's, then $b_1 = x_1 \otimes \mathrm{s}(\mathbf{f_2}) \otimes b_2 = x_2 \otimes \mathrm{s}(\mathbf{f_2}) \otimes b_2$, so by cancellation $x_1 = x_2$, so this choice is unique. Similarly, if they are right transposable they have a unique right transposition.

If they are both left and right transposable, then there exists x_1, x_2 such that

$$\begin{aligned} a_2 &= a_1 \otimes \mathrm{t}(\mathbf{f_1}) \otimes x_1 \\ b_1 &= x_1 \otimes \mathrm{s}(\mathbf{f_2}) \otimes b_2 \\ a_1 &= a_2 \otimes \mathrm{s}(\mathbf{f_2}) \otimes x_2 \\ b_2 &= x_2 \otimes \mathrm{t}(\mathbf{f_1}) \otimes b_1 \end{aligned}$$

This implies $\mathrm{t}(\mathbf{f_1}) = \mathrm{s}(\mathbf{f_2}) = x_1 = x_2 = \emptyset$, and $a_1 = a_2, b_1 = b_2$. But this is a contradiction to non-degeneracy. □

We note that being left transposable is equivalent to $L(f_1, f_2, g_1, g_2)$ holding. Similarly, being right transposable is equivalent to $L(g_1, g_2, f_1, f_2)$ holding. This gives us an alternative description of part of Theorem 2.16:

Theorem 2.28. *Two non-identity 1-molecules are equivalent if and only if one can be transformed into the other by a finite sequence of transpositions.*

Definition 2.29. Define ***i*-transposition** of 1-molecules to be the partial operation on 1-molecules of transposing the atoms at index i and $i + 1$; this is defined if and only if the 1-molecule is of length at least $i + 2$, and the designated atoms are transposable. Denote this operation by τ_i.

Definition 2.30. Let f, g be 1-molecules. We will call a finite list $(\tau_{k_i})_{i \in I}$ a **witness** for equivalence between f and g if the composite operation of applying them in order from left to right maps f to g. We will write $\mu : f \rightsquigarrow g$ to indicate μ is a witness mapping f to g, and $\mu^{-1} : g \rightsquigarrow f$ will be the corresponding witness mapping g to f.

Construction 2.31. If f and g are two equivalent 1-molecules, and T is a tag list on f, then any witness $\mu : f \rightsquigarrow g$ gives rise to an induced tag list T' on g by transposing in sequence the corresponding tags; that is, T' is the corresponding permutation of T. Reversing the witness sends T' back to T.

Definition 2.32. Say that two tagged 1-molecules f and g are **tag equivalent** if there exists a witness $\mu : f \rightsquigarrow g$ to their equivalence such that g's tag is induced by μ. Such a witness is called a **tag equivalence**.

We will show that all tag equivalences are uniquely determined by their underlying permutation, in an appropriate sense. To do so, we make use of the following fact (c.f. [BMMS00]):

Lemma 2.33. *For any symmetric group S_n, there exists a terminating and confluent rewriting system of the form:*

$$\sigma_i^2 \to 1 \qquad (i \geq 0)$$

$$\sigma_j\sigma_i \to \sigma_i\sigma_j \qquad (j \geq i+2)$$

$$\sigma_{i+k}\sigma_{i+k-1}\cdots\sigma_{i+1}\sigma_i\sigma_{i+k} \to \sigma_{i+k-1}\sigma_{i+k}\sigma_{i+k-1}\cdots\sigma_{i+1}\sigma_i \qquad (i \geq 0, k \geq 1)$$

where σ_i transposes i and $i+1$. Thus there exists a corresponding normal form for S_n in terms of transpositions, and every permutation expressed in transpositions can be reduced to normal form through an application of these rewriting rules. A term is in this normal form iff it is of the form $\prod_i C_i$ where each C_i is $\sigma_{a_i+N_i}\cdots\sigma_{a_i}$ and $a_i+N_i < a_j+N_j$ if $i<j$.

Note that the standard notation $\sigma_i\sigma_j$ for permutation groups, when considered as a left action on $\{1,\cdots,n\}$, corresponds to first applying σ_j and then applying σ_i. Our notation for a corresponding transposition sequence on 1-molecules would be (τ_j,τ_i), written in the reverse order

Lemma 2.34. *If $(\tau_{k_1},\ldots,\tau_{k_m})$ and $(\tau_{\ell_1},\ldots,\tau_{\ell_m})$ are transposition sequences with the same underlying permutation, then they define the same operation on 1-molecules.*

The proof of Lemma 2.34 is long but straight-forward; we defer it to Section 2.14.

Corollary 2.35. *Let f, g, g' be 1-molecules such that $\mu : f \rightsquigarrow g$ and $\nu : f \rightsquigarrow g'$. If μ and ν have the same underlying permutation, then $g = g'$.*

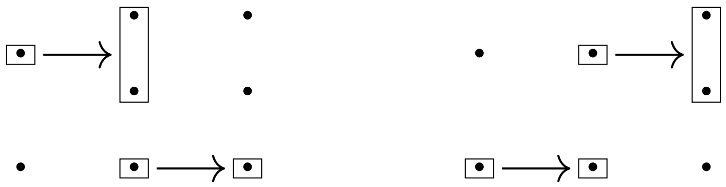

Figure 2.4: Transposition of two adjacent 1-atoms (without labels). Recall that we read tensors from top to bottom; thus the 1-atoms on the left are left transposable. The 1-atoms on the right are the result of their transposition.

Proof. By Lemma 2.34, $g = \mu(f) = \nu(f) = g'$. □

Corollary 2.36. *Let T and T' be contiguous tag lists partitioning a 1-molecule f, and let $f_T, f_{T'}$ denote the associated submolecules. Let $\mu : f \rightsquigarrow g$ be such that μ's underlying permutation does not exchange any elements of T with elements of T'. Then μ can be restricted to give witnesses $f_T \rightsquigarrow g_{\mu(T)}$ and $f_{T'} \rightsquigarrow g_{\mu(T')}$.*

Proof. A permutation which does not exchange elements of T and T' has a normal form which does not involve transpositions across their boundary; thus we may group the transpositions in the normal form into those affecting T and those affecting T'. These give the restricted witnesses. □

2.9 Interchangeability for 1-molecules

Definition 2.37. Two tags i, j in a tagged 1-molecule f are **interchangeable** if there are tag equivalent 1-molecules g, h such that $i \prec j$ in the tag list for g, and $j \prec i$ in the tag list for h. (One of g, h can be chosen to be f.)

Lemma 2.38. *Let f be a tagged 1-molecule, and i, j be tags. If i and j are interchangeable in f, and $\mu : f \rightsquigarrow g$, then i and j are interchangeable in g.*

Proof. This follows trivially from the definition. □

Lemma 2.39. *If $f \cong g$ are tagged 1-molecules and $i \prec j$ are adjacent tags transposable in f, then if $i \prec j$ in g and they are adjacent in g, they are transposable.*

Proof. Let $\mu : f \rightsquigarrow g$ be a normal form witness to $f \cong g$. Normal form witnesses transpose tags rightward until they reach their final spot, starting with the tag that ends rightmost; thus we may break μ into four contiguous transposition sequences, in order of application:

(i) μ_1 acts only on tags succeeding j in f.

(ii) μ_2 moves j rightward.

(iii) μ_3 moves i rightward

(iv) μ_4 moves tags preceding i in $(\mu_3 \circ \mu_2 \circ \mu_1)(f)$ rightward, potentially past both i and j. If it moves a tag past one, it moves it past both, since they end adjacent and in the same order.

Let k be the index of j, so that $\mu_2 = (\tau_k, \dots, \tau_{k+\ell})$ and $\mu_3 = (\tau_{k-1}, \dots, \tau_{k+\ell-1})$. We will show that, as partial operations, $(\mu \circ \tau_{k-1})(f) = (\tau_q \circ \mu)(f)$ for some q such that i is the tag at index q in $\mu(f)$. This implies that the operation of transposing i and j before applying μ will be the same as the operation of transposing after applying μ; therefore, since i and j are transposable in f they must be transposable in g. We'll make use of the fact that the composition operation $\mu_3 \circ \mu_2$ is also given by

$$(\tau_k, \tau_{k-1}, \tau_{k+1}, \tau_k, \dots, \tau_{k+\ell}, \tau_{k+\ell-1})$$

since this involves no additional transposability constraints and has the same underlying permutation.

Since μ_1 acts only on indices greater than k, it commutes with τ_{k-1}. Next note that

$$\mu_3 \circ \mu_2 \circ \tau_{k-1} = \tau_{k+\ell} \circ \mu_3 \circ \mu_2$$

by successive applications of the rule

$$(\tau_{k+i-1}, \tau_{k+i}, \tau_{k+i-1}) = (\tau_{k+i}, \tau_{k+i-1}, \tau_{k+i}) \tag{2.5}$$

from Lemma 2.74.

Next, we'll show that $\tau_{k+\ell}$ commutes appropriately with μ_4. Let $\mu_4 = C_0 \circ \cdots \circ C_M$ where $C_m = (\tau_{a_m}, \ldots, \tau_{a_m+N_m})$ as in Lemma 2.33; here each C_m is the rightward movement of a single tag. If $a_m + N_m < k + \ell - 1$, then it clearly commutes with $\tau_{k+\ell}$. Let n be the largest integer such that $a_n + N_n \geq k + \ell - 1$; then since applying the parts of μ up to this point puts i, j at indices $k + \ell, k + \ell + 1$, by construction we must have $a_n + N_n \geq k + \ell$ since it transposes past both. Then

$$\begin{aligned}(\tau_{k+\ell}, \tau_{a_n}, \ldots, \tau_{k+\ell-1}, \tau_{k+\ell}, \ldots, \tau_{a_n+N_n}) &= (\tau_{a_n}, \ldots, \tau_{k+\ell}, \tau_{k+\ell-1}, \tau_{k+\ell}, \ldots \tau_{a_n+N_n}) \\ &= (\tau_{a_n}, \ldots, \tau_{k+\ell-1}, \tau_{k+\ell}, \tau_{k+\ell-1}, \ldots, \tau_{a_n+N_n}) \\ &= (\tau_{a_n}, \ldots, \tau_{a_n+N_n}, \tau_{k+\ell-1})\end{aligned}$$

again by applying Eq. (2.5) and commuting past independent transpositions. Thus $C_n \circ \tau_{k+\ell} = \tau_{k+\ell-1} \circ C_n$. We can continue inductively to see $\mu_4 \circ \tau_{k+\ell} = \tau_{k+\ell-p} \circ \mu_4$ for some p, such that i is the tag at index $k + \ell - p$, as desired. □

Lemma 2.40. *Let f be a tagged 1-molecule. Two tags are interchangeable in f if and only if they are interchangeable in all submolecules which include the span of the tags.*

Proof. Let $i \prec j$ be interchangeable tags in f at indices p and q respectively. We will proceed by induction on $q - p$. If $q - p = 1$, then the tags are adjacent; since they are interchangeable, they are transposable in some equivalent molecule g. By Lemma 2.39, this implies they are transposable in f; thus they are interchangeable in the submolecule $f_p \star f_{p+1}$ as desired.

Now assume $q - p > 1$. Again, there is some molecule $g \cong f$ such that i and j are adjacent and transposable (else they could not interchange). For each tag m such that $i \prec m \prec j$ in f, m must be interchangeable with either i or j since it falls on one side or the other in g. Let ℓ be the tag at index $p+1$. If ℓ is transposable with i, τ_p is valid; let $\widetilde{f} = \tau_p(f)$. Apply the induction hypothesis to $\widetilde{f}$.

If ℓ is not transposable with i, it is not interchangeable with i by Lemma 2.39. Thus it must be interchangeable with j. Apply the inductive hypothesis to ℓ and j to get a transposition sequence μ interchanging ℓ and j by acting only on the submolecule which they span. Then in $\mu(f)$, the distance between i and j is smaller, so we can apply the induction hypothesis to get a transposition sequence ν interchanging them (starting from $\mu(f)$) by acting only on the submolecule which they span. Then $\mu \circ \nu$ interchanges i and j by acting on f. □

Definition 2.41. Let f be a tagged 1-molecule and let A, B be disjoint sets of tags in f. Then A and B are **interchangeable** in f if there exist $g \cong f, h \cong f$ such that all the tags of A precede all the tags of B in g, and all the tags of B precede all the tags of A in h.

Lemma 2.42. *If f is a tagged 1-molecule then A and B are interchangeable in f if and only if for all $a \in A$ and $b \in B$, a and b are interchangeable in f.*

Proof. The only if case is just the definition of interchangeability for tags, witnessed by the collective interchange of the sets A and B.

For the if case, we will show that f is tag equivalent to a 1-molecule with every tag of A to the left of every tag in B; the case where B is to the left is analogous. If f is already such a 1-molecule we are done. If not, let $a \in A$ be the leftmost tag in A such that some tag of B is to the left of it in f, and let $b \in B$ be the rightmost such tag. By assumption, a and b are interchangeable; then by Lemma 2.40, a and b can be interchanged in the submolecule spanning them. By construction, that submolecule doesn't contain any other tag of A or B. Thus we can apply a transposition sequence to f to move a to the left of b without impacting the relative positions of any other tags. Repeat until all tags of A are to the left. □

Definition 2.43. Two tag sets A and B in a 1-molecule f are **left interchangeable** if they are interchangeable, and whenever $i \in A$ and $j \in B$ are adjacent in an equivalent 1-molecule with $i \prec j$, then they are left transposable. Define **right interchangeable** similarly.

2.10 Recognition principle for 1-molecule decompositions

Theorem 2.44. *A 1-molecule f is equivalent to one of the form $g \otimes h$ for two 1-molecules $g, h \neq 1_\emptyset$ iff at least one of the following holds:*

(i) $f = 1_{a \otimes b}$ for 0-molecules $a, b \neq \emptyset$; in this case $g = 1_a$, $h = 1_b$.

(ii) Every 1-atom of f has a shared 0-molecule prefix a; in this case $g = 1_a$, and h is the 1-molecule coming from removing the prefix on each 1-atom.

(iii) Every 1-atom of f has a shared 0-molecule suffix b; in this case $h = 1_b$, and g is the 1-molecule coming from removing the suffix on each 1-atom.

(iv) f can be tagged and its tag set partitioned into two non-empty sets A and B such that A is left interchangeable with B.

Proof. ($\Rightarrow$) These can be seen directly from the definition of $\otimes$.
($\Leftarrow$) The implication is clear for (i) through (iii).

For (iv), choose a tag equivalent 1-molecule m such that A precedes B; let $p = \#A$ and $q = \#B$. Let A' and B' be the corresponding tag lists with the given ordering from m. Let T_i be the tag for index i in m, so that $A' = (T_0, \ldots, T_{p-1})$ and $B' = (T_p, \ldots, T_{p+q-1})$. Let $m_i = a_i \otimes \mathbf{m_i} \otimes b_i$. We will prove by induction on q that there exist 1-molecules g, h and a 0-molecule w such that $m = g \otimes 1_w \otimes h$, g does not have any shared 0-molecule suffix (as in case (iii)), and h does not have any shared 0-molecule prefix (as in case (ii)), and the induced tag list on m comes from tagging g with A and h with B.

Assume $q = 1$. Then by left transposability of T_{p-1} and T_p. there exists x_{p-1} such that $b_{p-1} = x_{p-1} \otimes \mathrm{s}(\mathbf{m_p}) \otimes b_p$. By left transposing T_{p-1} and T_p, we get a new 1-molecule $m^{(\tau_{p-1})}$

such that $m_{p-1}^{(\tau_{p-1})} = a_{p-1} \otimes \mathrm{s}\big(\mathbf{m_{p-1}}\big) \otimes x_{p-1} \otimes \mathbf{m_p} \otimes b_p$. Therefore, by left transposability of T_{p-2} and T_p, there exists x_{p-2} such that $b_{p-2} = x_{p-2} \otimes \mathrm{s}\big(\mathbf{m_p}\big) \otimes b_p$. Proceeding similarly, there exist x_i such that $b_i = x_i \otimes \mathrm{s}\big(\mathbf{m_p}\big) \otimes b_p$ for each $0 \leq i \leq p-1$. Hence by cancellation $a_i \otimes \mathrm{t}\big(\mathbf{m_i}\big) \otimes x_i = a_{i+1} \otimes \mathrm{s}\big(\mathbf{m_{i+1}}\big) \otimes x_{i+1}$ for $0 \leq i \leq p-2$. Let $\widetilde{g}_i = a_i \otimes \mathbf{m_i} \otimes x_i$, and let $\widetilde{g} = \widetilde{g}_0 \star \cdots \star \widetilde{g}_{p-1}$. Let w be the maximal shared 0-molecule suffix of $\widetilde{g}$, such that $\widetilde{g} = g \otimes 1_w$. Let $h = \mathbf{m_p} \otimes b_p$. Then $m = g \otimes 1_w \otimes h$ as desired.

Now assume $q > 1$. By the induction hypothesis, we have $m_0 \star \cdots \star m_{p+q-2} = g \otimes 1_w \otimes h \cong (1_{\mathrm{s}(g)} \otimes 1_w \otimes h) \star (g \otimes 1_w \otimes 1_{\mathrm{t}(h)})$. Here $g \otimes 1_w \otimes 1_{\mathrm{t}(h)}$ has induced tag list A'. Let $g \otimes 1_w \otimes 1_{\mathrm{t}(h)} = k_0 \star \cdots \star k_{p-1}$ where $k_i = a_i \otimes \mathbf{m_i} \otimes c_i \otimes w \otimes \mathrm{t}(h)$. Now consider $k_0 \star \cdots \star k_{p-1} \star m_{p+q-1}$; by reasoning as in the $q = 1$ case, we see that there exist x_i such that $c_i \otimes w \otimes \mathrm{t}(h) = x_i \otimes \mathrm{s}\big(\mathbf{m_p}\big) \otimes b_p$. We must have $\mathrm{length}(\mathrm{s}\big(\mathbf{m_p}\big) \otimes b_p) \leq \mathrm{length}(w \otimes \mathrm{t}(h))$ since the atoms of g have no shared suffix.

If $\mathrm{length}(\mathrm{s}\big(\mathbf{m_p}\big) \otimes b_p) \leq \mathrm{length}(\mathrm{t}(h))$, let u be such that $\mathrm{t}(h) = u \otimes \mathrm{s}\big(\mathbf{m_p}\big) \otimes b_p$. In this case, let $g' = g$, $w' = w$, and $h' = h \star (u \otimes \mathbf{m_p} \otimes b_p)$. If $\mathrm{length}(\mathrm{s}\big(\mathbf{m_p}\big) \otimes b_p) > \mathrm{length}(\mathrm{t}(h))$, let y, z be such that $w = y \otimes z$ and $z \otimes \mathrm{t}(h) = \mathrm{s}\big(\mathbf{m_p}\big) \otimes b_p$. In this case, let $g' = g$, $w' = y$, $h' = (1_z \otimes h) \star (\mathbf{m_p} \otimes b_p)$. In either case, $m = g' \otimes 1_{w'} \otimes h'$ as desired. This finishes the induction.

We may turn a decomposition $g \otimes 1_w \otimes h$ into a binary decomposition by taking any partition $w = w' \otimes w''$, so that $f \cong (g \otimes 1_{w'}) \otimes (1_{w''} \otimes h)$. □

We can note that the above proof actually yields a list of all possible decompositions (up to equivalence). Case (i) shows that we can $\otimes$-decompose identity 1-molecules by partitioning their representing 0-molecule. Cases (ii) and (iii) show us that we can decompose as a product of an identity and non-identity molecule by checking for shared prefixes/suffixes. To construct all decompositions for case (iv), we can first find all partitions A and B satisfying the interchange property, then use the proof procedure to find g, w, h. As illustrated, any partition of w gives a binary decomposition; it's straightforward to show there's no other way to write m as a $\otimes$ of two non-trivial 1-molecules of lengths $|A|$ and

$|B|$. If we choose some tag-equivalent molecule κ s.t. $A \prec B$, then m is equivalent to κ by re-ordering A and B separately; thus $m_0 \star \cdots \star m_{p-1} \cong \kappa_0 \star \cdots \star \kappa_{p-1}$. Such a re-ordering does not affect any shared 0-molecule suffix or prefix, so induces equivalences $g_m \cong g_\kappa$ and $h_m \cong h_\kappa$. Thus by re-ordering we get an equivalent decomposition.

2.11 Cancellation for 1-molecules

Lemma 2.45. *Two adjacent 1-atoms can not be both left and right transposable.*

Proof. This is shown in the proof of Lemma 2.27. □

Lemma 2.46. *Let $\alpha, \beta, \alpha', \beta', \alpha'', \beta''$ be 1-atoms such that $\tau_0(\alpha \star \beta') = \beta \star \alpha'$, $\tau_0(\alpha \star \beta'') = \beta \star \alpha''$. Then $\alpha' = \alpha''$ and $\beta' = \beta''$.*

Proof. Let $\alpha = a_\alpha \otimes \mathbf{p}_\alpha \otimes b_\alpha$, and similarly for each of the other 1-atoms. Note that since transposition preserves the primitive of an atom, $\mathbf{p}_\alpha = \mathbf{p}_{\alpha'} = \mathbf{p}_{\alpha''}$ and $\mathbf{p}_\beta = \mathbf{p}_{\beta'} = \mathbf{p}_{\beta''}$.

We will break up into cases based on the type of each transposition. If both transpositions are left transpositions, then by Lemma 2.26 there exists x_1, x_2 such that $\alpha = a_\alpha \otimes \mathbf{p}_\alpha \otimes x_1 \otimes \mathrm{s}(\mathbf{p}_\beta) \otimes b_{\beta'} = a_\alpha \otimes \mathbf{p}_\alpha \otimes x_2 \otimes \mathrm{s}(\mathbf{p}_\beta) \otimes b_{\beta''}$, so $b_\alpha = x_1 \otimes \mathrm{s}(\mathbf{p}_\beta) \otimes b_{\beta'} = x_2 \otimes \mathrm{s}(\mathbf{p}_\beta) \otimes b_{\beta''}$. Thus, by the action of transposition in each case $\beta = a_\alpha \otimes \mathrm{s}(\mathbf{p}_\alpha) \otimes x_1 \otimes \mathbf{p}_\beta \otimes b_{\beta'} = a_\alpha \otimes \mathrm{s}(\mathbf{p}_\alpha) \otimes x_2 \otimes \mathbf{p}_\beta \otimes b_{\beta''}$. Thus $b_{\beta'} = b_{\beta''}$ and $x_1 = x_2$. Since $\beta' = a_\alpha \otimes \mathrm{t}(\mathbf{p}_\alpha) \otimes x_1 \otimes \mathbf{p}_\beta \otimes b_{\beta'}$ and $\beta'' = a_\alpha \otimes \mathrm{t}(\mathbf{p}_\alpha) \otimes x_2 \otimes \mathbf{p}_\beta \otimes b_{\beta''}$, we have $\beta' = \beta''$. Similarly, $\alpha' = \alpha''$. The case where both are right transpositions is similar.

If the first transposition is a left transposition, and the second is a right transposition, then again by Lemma 2.26 there exists x_1, x_2 such that

$$\begin{aligned}
\alpha &= a_\alpha \otimes \mathbf{p}_\alpha \otimes x_1 \otimes \mathrm{s}(\mathbf{p}_\beta) \otimes b_{\beta'} \\
&= a_{\beta''} \otimes \mathrm{s}(\mathbf{p}_\beta) \otimes x_2 \otimes \mathbf{p}_\alpha \otimes b_\alpha \\
\beta' &= a_\alpha \otimes \mathrm{t}(\mathbf{p}_\alpha) \otimes x_1 \otimes \mathbf{p}_\beta \otimes b_{\beta'} \\
\beta'' &= a_{\beta''} \otimes \mathbf{p}_\beta \otimes x_2 \otimes \mathrm{t}(\mathbf{p}_\alpha) \otimes b_\alpha
\end{aligned}$$

By the action of transposition in each case,

$$\begin{aligned}\beta &= a_\alpha \otimes \mathrm{s}\left(\mathbf{p}_\alpha\right) \otimes x_1 \otimes \mathbf{p}_\beta \otimes b_{\beta'} \\ &= a_{\beta''} \otimes \mathbf{p}_\beta \otimes x_2 \otimes \mathrm{s}\left(\mathbf{p}_\alpha\right) \otimes b_\alpha\end{aligned}$$

Thus

$$\begin{aligned}a_\alpha &= a_{\beta''} \otimes \mathrm{s}\left(\mathbf{p}_\beta\right) \otimes x_2 \\ b_\alpha &= x_1 \otimes \mathrm{s}\left(\mathbf{p}_\beta\right) \otimes b_{\beta'} \\ a_{\beta''} &= a_\alpha \otimes \mathrm{s}\left(\mathbf{p}_\alpha\right) \otimes x_1 \\ b_{\beta'} &= x_2 \otimes \mathrm{s}\left(\mathbf{p}_\alpha\right) \otimes b_\alpha\end{aligned}$$

By substitution in the above,

$$\begin{aligned}a_\alpha &= a_\alpha \otimes \mathrm{s}\left(\mathbf{p}_\alpha\right) \otimes x_1 \otimes \mathrm{s}\left(\mathbf{p}_\beta\right) \otimes x_2 \\ b_\alpha &= x_1 \otimes \mathrm{s}\left(\mathbf{p}_\beta\right) \otimes x_2 \otimes \mathrm{s}\left(\mathbf{p}_\alpha\right) \otimes b_\alpha\end{aligned}$$

Thus $x_1, x_2, \mathrm{s}\left(\mathbf{p}_\alpha\right), \mathrm{s}\left(\mathbf{p}_\beta\right)$ must all be $\emptyset$. This is a contradiction to non-degeneracy, so this case can not occur. □

Lemma 2.47. *Let $F = \alpha \star f'$, $\widetilde{F} = f \star \alpha'$, $G = \alpha \star f''$, and $\widetilde{G} = f \star \alpha''$ be 1-molecules, where f, f', f'' are 1-molecules and $\alpha, \alpha', \alpha''$ are 1-atoms. Let α have tag a and f', f'' have tag list B such that F, G have tag list aB induced by composition. If $\mu : F \rightsquigarrow \widetilde{F}$ and $\nu : G \rightsquigarrow \widetilde{G}$ both have underlying permutation $aB \to Ba$, then $\alpha' = \alpha''$ and $f' = f''$.*

Proof. If any of the three of f, f', f'' is an identity 1-molecule, then the rest must also be by consideration of length; thus $f' = f'' = 1_{\mathrm{t}(\alpha)}$, and $\alpha = \alpha' = \alpha''$. Thus we can assume $f = f_0 \star \cdots \star f_\ell$, $f' = f'_0 \star \cdots \star f'_\ell$, $f'' = f''_0 \star \cdots \star f''_\ell$.

The normal form representative of μ and ν is $(\tau_0, \ldots, \tau_\ell)$; thus this is a valid transposition sequence on each of F and G. Observe that applying τ_0 to each of F and G, there exist 1-atoms α'_1, α''_1 s.t. we have equivalence of submolecules

$$\alpha \star f'_0 \cong f_0 \star \alpha'_1$$

$$\alpha \star f''_0 \cong f_0 \star \alpha''_1$$

Thus, by Lemma 2.46 we have $f'_0 = f''_0$ and $\alpha'_1 = \alpha''_1$. We may proceed inductively, ending with $\alpha' = \alpha'_\ell = \alpha''_\ell = \alpha''$ by Lemma 2.34. Note that $f'_0 \star \cdots \star f'_\ell = f''_0 \star \cdots \star f''_\ell$ – they are not just equivalent. □

Lemma 2.48. *Let $F = f \star g'$, $\widetilde{F} = g \star f'$, $G = f \star g''$, and $\widetilde{G} = g \star f''$ be 1-molecules such that* $\text{length}(f) = \text{length}(f') = \text{length}(f'')$ *and* $\text{length}(g) = \text{length}(g') = \text{length}(g'')$. *Let f have tag list A and g', g'' have tag list B, such that F, G have the tag list AB induced by composition. If $\mu : F \rightsquigarrow \widetilde{F}$ and $\nu : G \rightsquigarrow \widetilde{G}$ both have underlying permutation of block transposition $AB \to BA$, then $f' = f''$ and $g' = g''$.*

Proof. If any of f, f', f'' are identity molecules, then the rest must be since they have the same length. In this case, μ and ν are the identity so $g = g' = g''$, and $f' = 1_{\mathsf{t}(g)} = f''$. Similarly if any of g, g', g'' are the identity molecule. Thus we can assume they are both non-identities.

Let $\ell = \text{length}(f)$ and $k = \text{length}(g)$. Since A and B are interchangeable in F and G, there exist 1-atoms $\widetilde{g}'_0, \widetilde{g}''_0, f'_{(0,i)}$, and $f''_{(0,i)}$ for $0 \leq i \leq \ell - 1$ such that the transposition sequence $(\tau_{\ell-1}, \cdots, \tau_0)$ when applied to initial submolecules of F and G yields

$$f_0 \star \cdots \star f_{\ell-1} \star g'_0 \cong \widetilde{g}'_0 \star f'_{(0,0)} \star \cdots \star f'_{(0,\ell-1)}$$

$$f_0 \star \cdots \star f_{\ell-1} \star g''_0 \cong \widetilde{g}''_0 \star f''_{(0,0)} \star \cdots \star f'_{(0,\ell-1)}$$

We can continue similarly applying $(\tau_{\ell-1+i}, \cdots, \tau_i)$ in order for increasing i to show that there exist 1-atoms such that

$$f'_{(i,0)} \star \cdots \star f'_{(i,\ell-1)} \star g'_{i+1} \cong \widetilde{g}'_{i+1} \star f'_{(i+1,0)} \star \cdots \star f'_{(i+1,\ell-1)}$$

$$f''_{(i,0)} \star \cdots \star f''_{(i,\ell-1)} \star g''_{i+1} \cong \widetilde{g}''_{i+1} \star f''_{(i+1,0)} \star \cdots \star f''_{(i+1,\ell-1)}$$

The transposition sequence given by the concatenation of all of these has the same underlying permutation as μ and ν; thus by Lemma 2.34, it represents the same operation on 1-molecules. Thus $\widetilde{g}'_i = \widetilde{g}''_i = g_i$ for all i. Additionally, $f'_{(k-1,j)} = f'_j$ and $f''_{(k-1,j)} = f''_j$ for all j. Therefore

$$f_0 \star \cdots \star f_{\ell-1} \star g'_0 \cong g_0 \star f'_{(0,0)} \star \cdots \star f'_{(0,\ell-1)}$$

$$f_0 \star \cdots \star f_{\ell-1} \star g''_0 \cong g_0 \star f''_{(0,0)} \star \cdots \star f'_{(0,\ell-1)}$$

Thus we can apply Lemma 2.47 to show $g'_0 = g''_0$ and $f'_{(0,j)} = f''_{(0,j)}$. Continuing inductively, we can show $g'_i = g''_i$ for each i and $f'_{(i,j)} = f''_{(i,j)}$, thus $f'_j = f''_j$, completing the proof. □

Lemma 2.49. *If f, g, h are 1-molecules such that $\pi : f \star g \rightsquigarrow h$, then there exists $\mu : f' \rightsquigarrow f, \nu : g' \rightsquigarrow g$ such that, if we tag f' and g' by disjoint tag lists A and B respectively, then $\pi \circ (\mu \star \nu) : f' \star g' \rightsquigarrow h$ induces a shuffle on A, B.*

Proof. Tag f and g by disjoint tag lists C and D, and let E be the tag list induced on h. Let A be the restriction of E to tags in C, and B the restriction of E to tags in D; thus A is a reordering of C and B is a re-ordering of D, and E is an (A, B)-shuffle. Let η be a normal form transposition sequence representing the reordering $C \rightsquigarrow A$ as in Lemma 2.33; an examination of normal form shows that it does not transpose any tags which its underlying permutation does not interchange. If η interchanges any tags in C, then π also interchanges those tags by construction. Thus the tags are interchangeable in $f \star g$, and therefore in f by Lemma 2.40. Therefore η is a valid transposition sequence on f by Lemma 2.39. Let ϵ similarly be a normal form transposition sequence representing the reordering $D \rightsquigarrow B$. Let $f' = \eta(f)$, $g' = \epsilon(g)$, $\mu = \eta^{-1}$, and $\nu = \epsilon^{-1}$. □

Lemma 2.50. *If f, g, h, k are 1-molecules such that $\pi : f \star g \Rrightarrow h \star k$, then there exist $\mu : \widetilde{f} \Rrightarrow f$, $\nu : \widetilde{g} \Rrightarrow g$, $\sigma : h \Rrightarrow \widetilde{h}$, and $\tau : k \Rrightarrow \widetilde{k}$ such that $\widetilde{f} = \widetilde{f}^0 \star \widetilde{f}^1, \widetilde{g} = \widetilde{g}^0 \star \widetilde{g}^1$, and if we tag $\widetilde{f}^0, \widetilde{f}^1, \widetilde{g}^0$, and $\widetilde{g}^1$ by disjoint tag lists A, B, C, and D, then $(\sigma \star \tau)\circ\pi\circ(\mu \star \nu)$ has underlying permutation $ABCD \Rrightarrow ACBD$.*

Proof. First, apply Lemma 2.49 to get μ, ν and tag lists E, F such that $\pi\circ(\mu\star\nu) : \widetilde{f}\star\widetilde{g} \Rrightarrow h \star k$ is an (E, F)-shuffle. Let $\Pi = \pi\circ(\mu \star \nu)$. Let A be the ordered list of tags $s \in E$ such that the index of $\Pi(s)$ is less than length(h). Similarly, let D be the ordered list of tags in $t \in F$ such that the index of $\Pi(t)$ is greater than or equal to length(h). Then by the shuffle property A is a prefix of E and D is a suffix of F, so let B, C be such that $E = AB$, $F = CD$. Define $\widetilde{f}^0, \widetilde{f}^1, \widetilde{g}^0$, and $\widetilde{g}^1$ to be the submolecules corresponding to tag lists A, B, C, and D .

Since Π is an (AB, CD)-shuffle, every tag in A is interchangeable in $h\star k$ with the tags to the left of it, and thus is interchangeable in h with those tags by Lemma 2.40. Additionally, the tags landing at indices less than length(h) come from A and C by construction, so the tag list of h is an (A, C)-shuffle. Define $\sigma : h \Rrightarrow \widetilde{h}$ by successively transposing the tags in A leftward, so that $\widetilde{h}$'s induced tag list is AC. Similarly, define $\tau : k \Rrightarrow \widetilde{k}$ by transposing the tags in D rightward, so that $\widetilde{k}$'s induced tag list is BD. Then $(\sigma \star \tau)\circ\Pi$ has the desired property. □

Theorem 2.51. *If f, g, x are 1-molecules such that $x \star f \cong x \star g$, then $f \cong g$. Similarly, if $f \star x \cong g \star x$ then $f \cong g$.*

Proof. We will only prove the first statement, since the proof of the second is analogous. Proceed by induction on length(x). If x is an identity 1-molecule, then the statement is trivially true.

Assume length$(x) > 0$. Let $\pi : x \star f \Rrightarrow x \star g$, and apply Lemma 2.50 to get $\mu : y \Rrightarrow x$, $\nu : \widetilde{f} \Rrightarrow f$, $\sigma : x \Rrightarrow z$, $\tau : g \Rrightarrow \widetilde{g}$ such that $y = \alpha \star \beta$ and $\widetilde{f} = \widetilde{f}^0 \star \delta$. By the lemma, if we tag $\alpha, \beta, \widetilde{f}^0, \delta$ by A, B, C, D, then $\Omega := (\sigma \star \tau)\circ\pi\circ(\mu \star \nu)$ has underlying permutation $ABCD \rightsquigarrow ACBD$. Thus, we know that there exists $\gamma, \widetilde{g}^0$ such that $z = \alpha \star \gamma$ and $\widetilde{g} = \widetilde{g}^0 \star \delta$. So $\Omega : \alpha \star \beta \star \widetilde{f}^0 \star \delta \rightsquigarrow \alpha \star \gamma \star \widetilde{g}^0 \star \delta$. By Corollary 2.36, restricting to the submolecule corresponding to ABC on the lefthand side, we have $\Omega' : \alpha \star \beta \star \widetilde{f}^0 \rightsquigarrow \alpha \star \gamma \star \widetilde{g}^0$. If we could prove $\widetilde{f}^0 \cong \widetilde{g}^0$, we would be done, since this would imply $f \cong \widetilde{f} \cong \widetilde{f}^0 \star \delta \cong \widetilde{g}^0 \star \delta \cong \widetilde{g} \cong g$.

If $\mathrm{length}(\alpha) = \mathrm{length}(x)$, then β, γ are identity molecules since $x \cong \alpha \star \beta \cong \alpha \star \gamma$. Thus $|B| = |C| = 0$, so we must have $\mathrm{length}(\widetilde{f}^0) = \mathrm{length}(\widetilde{g}^0) = 0$. Hence $\widetilde{f}^0 = \widetilde{g}^0 = 1_{\mathrm{t}(\alpha)}$.

If $\mathrm{length}(\alpha) < \mathrm{length}(x)$, apply the inductive hypothesis to $\alpha \star \beta \cong \alpha \star \gamma$ to see that $\beta \cong \gamma$; let ρ be a transposition sequence mapping β to γ. Restricting Ω' further to the tag list BC on the lefthand side, we have $\Omega'' : \beta \star \widetilde{f}^0 \rightsquigarrow \gamma \star \widetilde{g}^0$. The interchangeability relations in β are the same as in $\widetilde{g}^0$, since both have tag list B under the tag equivalence. Thus ρ is also a valid transposition sequence on $\widetilde{g}^0$. Let $\widetilde{g}^1 = \rho(\widetilde{g}^0)$ and $\Delta = (1 \star \rho) \circ \Omega'' \circ (\rho^{-1} \star 1)$. Then $\Delta : \gamma \star \widetilde{f}^0 \rightsquigarrow \gamma \star \widetilde{g}^1$ has underlying permutation $\rho(B)C \to C\rho(B)$. Apply Lemma 2.48 with $f = g = \gamma$, $g' = f'' = \widetilde{f}^0$, and $g'' = f' = \widetilde{g}^1$ and equivalences given by Δ and Δ^{-1} to get that $\widetilde{f}^0 = \widetilde{g}^1 \cong \widetilde{g}^0$ as desired. □

Corollary 2.52. *Let f, g, a, b, c, d be 1-molecules such that $f \star a \cong g \star b$ and $f \star c \cong g \star d$. Then there exists $\alpha, \beta, a', b', c', d'$ s.t. $f \star \alpha \cong g \star \beta$ and $a \cong \alpha \star a'$, $b \cong \beta \star b'$, $c \cong \alpha \star c'$, $d \cong \beta \star d'$.*

Proof. We will proceed by induction on the minimum length of f and g. If $f = 1_x$, then choose $\alpha = g, \beta = 1_{\mathrm{t}(g)}, a' = b, b' = b, c' = d, d' = d$. Proceed similarly if $g = 1_y$.

Now assume the minimum length is $n > 0$. Apply Lemma 2.50 to a witness $\pi_1 : f \star a \rightsquigarrow g \star b$ to get $\mu_1 : \widetilde{f} \rightsquigarrow f$, $\nu_1 : \widetilde{a} \rightsquigarrow a$, $\sigma_1 : g \rightsquigarrow \widetilde{g}$, and $\tau_1 : b \rightsquigarrow \widetilde{b}$ such that $\widetilde{f} = \widetilde{f}^0 \star \widetilde{f}^1$ and $\widetilde{a} = \widetilde{a}^0 \star \widetilde{a}^1$. Let $\Pi_1 = (\sigma_1 \star \tau_1) \circ \pi_1 \circ (\mu_1 \star \nu_1)$. Given that we tag $\widetilde{f}^0$, $\widetilde{f}^1$, $\widetilde{a}^0$, and $\widetilde{a}^1$ with disjoint tag lists A, B, C, and D, we additionally know from the lemma that Π_1 has underlying permutation $ABCD \rightsquigarrow ACBD$. Thus we can restrict it to $\widetilde{f}^0 \star \widetilde{f}^1 \star \widetilde{a}^0$, and get that $\widetilde{f} \star \widetilde{a}^0 \cong \widetilde{g} \star \widetilde{b}^0$ for some $\widetilde{b}^0$.

Given $\alpha', \beta', a'', b'', c'', d''$ such that $\widetilde{a}^0 \cong \alpha' \star a''$, $\widetilde{b}^0 \cong \beta' \star b''$, $c \cong \alpha' \star c''$, $d \cong \beta' \star d''$, and $f \star \alpha' \cong f \star \beta'$, we could take $\alpha = \alpha', \beta = \beta', a' = a'' \star \widetilde{a}^1, b' = b'' \star \widetilde{b}^1, c' = c''$, and $d' = d''$. Thus, it is sufficient to prove the result assuming that D is empty.

We will perform a similar decomposition on a witness $\pi_2 : f \star c \cong g \star d$. Let E, F, G, and H be disjoint tag lists also disjoint from A,B,C, and D. Define $\mu_2, \nu_2, \sigma_2, \tau_2, \overline{f}, \overline{f}^0, \overline{f}^1, \overline{c}^0, \overline{c}^1$, and Π_2 analogously to above. We can similarly reduce to the case where H is empty.

Assume without loss of generality that length(f) $\leq$ length(g). Since Π_1 has underlying permutation $ABC \to ACB$, both $\widetilde{f} \star a$ and $\widetilde{g} \star b$ share the prefix $\widetilde{f}^0$; therefore by length, both $\widetilde{f}$ and $\widetilde{g}$ share this prefix. If A is non-empty and thus $\widetilde{f}^0$ is not an identity molecule, then there exists $\widetilde{g}^1$ such that $g \cong \widetilde{f}^0 \star \widetilde{g}^1$. Since $f \cong \widetilde{f}^0 \star \widetilde{f}^1$, we can apply Theorem 2.51 to cancel in the equivalences $\widetilde{f}^0 \star \widetilde{f}^1 \star a \cong \widetilde{f}^0 \star \widetilde{g}^1 \star b$ and $\widetilde{f}^0 \star \widetilde{f}^1 \star c \cong \widetilde{f}^0 \star \widetilde{g}^1 \star d$. Applying the inductive hypothesis to $\widetilde{f}^1 \star a \cong \widetilde{g}^1 \star b$ and $\widetilde{f}^1 \star c \cong \widetilde{g}^1 \star d$, we get $\alpha, \beta, a', b', c', d'$ which also work as solutions to the original system. We may apply similar reasoning in the case that E is non-empty.

Thus we are left with the case where Π_1 and Π_2 have underlying permutations $BC \to CB$ and $FG \to GF$ respectively. These are in fact the same permutation, since length(f) = length(B) = length(F) and length(g) = length(C) = length(G). By Lemma 2.48, we must have $a = c$ and $b = d$. Thus choose $\alpha = a = c, \beta = b = d$, and the rest to be the appropriate identity molecules. □

Corollary 2.53. *Let f, g be 1-molecules such that there exists 1-molecules a, b with $f \star a \cong g \star b$. Let α, β be minimal length choices of a and b; then for any other minimal length choices α', β' we have $\alpha \cong \alpha'$ and $\beta \cong \beta'$. For any choice of a, b, we have $a \cong \alpha \star a'$, $b \cong \beta \star b'$.*

Proof. Note that α is of minimal length if and only if β is, since f and g are fixed. Given two choices α, β and α', β', apply Corollary 2.52 to $f \star \alpha \cong g \star \beta$ and $f \star \alpha' \cong g \star \beta'$. By minimality, this gives $\alpha \cong \alpha'$ and $\beta \cong \beta'$. For any other choice of a, b such that $f \star a \cong g \star b$, we can apply Corollary 2.52 on $f \star a \cong g \star b$ and $f \star \alpha \cong g \star \beta$ to get that $a \cong \alpha \star a'$ and $b \cong \beta \star b'$. □

2.12 Equivalence checking in Dimension 2

Definition 2.54. Let $x = l \star (a \otimes \mathbf{m} \otimes b) \star r$ be a 2-atom. Tag the collapse $\overline{x}$ and let t be the tag associated with the atom containing the prim $\overline{\mathbf{m}}$ (which occurs only once in the collapse by construction, since it is a primitive not in the original computad). We say that a 2-atom is in **righthand form** if t is to the left of every tag it's interchangeable with in $\overline{x}$. We say that it is in **lefthand form** if t is to the right of every tag it's interchangeable with.

Unlike 1-atoms, where equivalence and equality are the same, there may be multiple distinct 2-atoms representing the same 2-cell; from [Mak05], two 2-atoms represent the same 2-cell if they share the same primitive and their collapses are equivalent 1-molecules in the corresponding strict monoidal computad. Note that all equivalent 2-atoms share the same 2-primitive, and every 2-atom has a (non-canonical) lefthand and righthand form.

Definition 2.55. If two non-identity 2-molecules have the same number of atoms and their atoms are pairwise equivalent, call the 2-molecules **atom-equivalent**. Call two identity 2-molecules atom-equivalent if their base 1-molecules are equivalent. Denote the equivalence class of a 2-atom α by $[\alpha]$, and the atom-equivalence class of a 2-molecule f by $[f]$. Call such an equivalence class an **AE-2-molecule**.

Lemma 2.56. *If $x_i = l_i \star (a_i \otimes \mathbf{m_i} \otimes b_i) \star r_i$ for $i = 1, 2$ are two 2-atoms in righthand form, then they are equivalent if and only if $\mathbf{m_1} = \mathbf{m_2}$, $a_1 = a_2$, $b_1 = b_2$, $l_1 \cong l_2$, and $r_1 \cong r_2$. (The same result holds for lefthand form.)*

Proof. ($\Leftarrow$) We can extend and compose the witnesses of equivalence on l_i and r_i to a witness that the collapses are equivalent.

($\Rightarrow$) Let $\mathbf{m} := \mathbf{m_1} = \mathbf{m_2}$. Let $\mu : \overline{x_1} \rightsquigarrow \overline{x_2}$. Tag $\overline{x_1}$, and let t be the tag associated to the atom whose primitive is $\overline{\mathbf{m}}$. Let A be the tag list to the left of t and B the tag list to the right, such that the tag list of $\overline{x_1}$ is AtB. Similarly, let the tag list induced on $\overline{x_2}$ by μ be CtD.

Since each was in righthand form, none of the tags in A or C are interchangeable with t. Thus the underlying permutation of μ acts independently on A – if $\mathtt{length}(A) = k$, it can't map an element of B into the first k spots, since such an element would end up to the left of t but be interchangeable with it, a contradiction. Therefore by Corollary 2.36, μ restricts to give $\eta : l_1 \rightsquigarrow l_2$. Now since μ restricted to tB has underlying permutation $tB \rightsquigarrow tD$, it also restricts to give $\epsilon : r_1 \rightsquigarrow r_2$. Since μ acts as the identity when restricted to t, we must have $a_1 = a_2, b_1 = b_2$ as desired. □

Definition 2.57. Let $f_i = l_i \star (a_i \otimes \mathbf{f_i} \otimes b_i) \star r_i$ for $i = 1, 2$ be adjacent 2-atoms in a 2-molecule. These are said to be **left transposable** if there exist 2-atoms α, β such that $f_1 \cong \alpha \star \mathrm{s}(\beta)$ and $f_2 \cong \mathrm{t}(\alpha) \star \beta$ (as 2-atoms). Their (α, β) **left transposition** is then the pair of adjacent 2-atoms $\mathrm{s}(\alpha) \star \beta$ and $\alpha \star \mathrm{t}(\beta)$. They are said to be **right transposable** if there exist 2-atoms α, β such that $f_1 \cong \mathrm{s}(\alpha) \star \beta$ and $f_2 \cong \alpha \star \mathrm{t}(\beta)$. Their (α, β) **right transposition** is then the pair of adjacent 2-atoms $\alpha \star \mathrm{s}(\beta)$ and $\mathrm{t}(\alpha) \star \beta$.

Construction 2.58. Let $f_i = l_i \star (a_i \otimes \mathbf{f_i} \otimes b_i) \star r_i$ be a 2-atom for $i = 1, 2$. If f_1 is in righthand form and f_2 is in lefthand form then for every selection of α, β where α is in righthand form, β is in lefthand form, and they give a left transposition of f_1, f_2, we can construct a 1-molecule $x_{(\alpha,\beta)}$ such that

$$\begin{aligned} l_2 &\cong l_1 \star (a_1 \otimes \mathrm{t}\big(\mathbf{f_1}\big) \otimes b_1) \star x_{(\alpha,\beta)} \\ r_1 &\cong x_{(\alpha,\beta)} \star (a_2 \otimes \mathrm{s}\big(\mathbf{f_2}\big) \otimes b_2) \star r_2 \end{aligned} \tag{2.6}$$

as follows. By Lemma 2.56 and the definition of left transposability, we must have:

$$\begin{aligned} &a_1 = a_\alpha && b_1 = b_\alpha \\ &a_2 = a_\beta && b_2 = b_\beta \\ &l_1 \cong l_\alpha && r_1 \cong r_\alpha \star l_\beta \star (a_\beta \otimes \mathrm{s}\big(\mathbf{f}_\beta\big) \otimes b_\beta) \star r_\beta \\ &l_2 \cong l_\alpha \star (a_\alpha \otimes \mathrm{t}\big(\mathbf{f}_\alpha\big) \otimes b_\alpha) \star r_\alpha \star l_\beta && r_2 \cong r_\beta \end{aligned} \tag{2.7}$$

Thus we take $x_{(\alpha,\beta)} = r_\alpha \star l_\beta$. If $\alpha \cong \alpha'$, $\beta \cong \beta'$ and α' is also in righthand form, β' in lefthand form, then $r_{\alpha'} \cong r_\alpha$, $l_{\beta'} \cong l_\beta$. Thus $[x_{(\alpha,\beta)}] = [x_{(\alpha',\beta')}]$.

For any α, β (not necessarily in righthand/lefthand form), we can thus define a unique 1-molecule equivalence class $X_{(\alpha,\beta)}$ by taking $[x_{\alpha',\beta'}]$ for any selection α', β' of righthand and lefthand forms respectively..

The same construction works if f_1, α in lefthand form and f_2, β in righthand form, except with the conditions

$$l_1 \cong l_2 \star (a_2 \otimes \mathrm{s}(\mathbf{f_2}) \otimes b_2) \star x_{(\alpha,\beta)}$$
$$r_2 \cong x_{(\alpha,\beta)} \star (a_1 \otimes \mathrm{t}(\mathbf{f_1}) \otimes b_1) \star r_1$$

Lemma 2.59. *Let $f_i = l_i \star (a_i \otimes \mathbf{f_i} \otimes b_i) \star r_i$ for $i = 1, 2$ be adjacent 2-atoms with f_1 in righthand form and f_2 in lefthand form. They are left transposable if and only if there exists x such that*

$$\begin{aligned} l_2 &\cong l_1 \star (a_1 \otimes \mathrm{t}(\mathbf{f_1}) \otimes b_1) \star x \\ r_1 &\cong x \star (a_2 \otimes \mathrm{s}(\mathbf{f_2}) \otimes b_2) \star r_2 \end{aligned} \tag{2.8}$$

Additionally, all such x are equivalent. Similarly, the f_i are right transposable if and only if there exists x such that

$$l_1 \cong l_2 \star (a_2 \otimes \mathrm{s}(\mathbf{f_2}) \otimes b_2) \star x$$
$$r_2 \cong x_\star (a_1 \otimes \mathrm{t}(\mathbf{f_1}) \otimes b_1) \star r_1$$

Proof. If the f_i are left transposable, then for any choice of α, β giving such a transposition we can take any $x \in X_{(\alpha,\beta)}$.

Given a 1-molecule x satisfying Eq. (2.8), define

$$\alpha_x = l_1 \star (a_1 \otimes \mathbf{f_1} \otimes b_1) \star 1_{\mathrm{s}(x)}$$
$$\beta_x = x \star (a_2 \otimes \mathbf{f_2} \otimes b_2) \star r_2$$

Then α_x is in righthand form and β_x is in lefthand form, and they give a left transposition of f_1, f_2 such that $x_{(\alpha_x,\beta_x)} = x$. It remains to show that all such x are equivalent. If x_1, x_2 both satisfied the conditions in Eq. (2.8), then we'd have $l_2 \cong l_1 \star (a_1 \otimes \mathrm{t}(\mathbf{f_1}) \otimes b_1) \star x_1 \cong l_1 \star (a_1 \otimes \mathrm{t}(\mathbf{f_1}) \otimes b_1) \star x_2$. By Theorem 2.51, $x_1 \cong x_2$ as desired. □

Lemma 2.60. *If f and g are left transposable 2-atoms, they have a unique left transposition up to equivalence of 2-atoms. The same holds for right transpositions.*

Proof. We prove only for the left transposition case. If f and g are left transposable, then we may assume f in righthand form and g in lefthand form since these will have the same left transpositions. Then for every α, β giving a left transposition, any $x_{(\alpha',\beta')} \in X_{(\alpha,\beta)}$ satisfies the conditions of Lemma 2.59. Therefore, if γ, δ is another choice giving a left transposition and $x_{(\gamma',\delta')} \in X_{(\gamma,\delta)}$ then $x_{(\alpha',\beta')} \cong x_{(\gamma',\delta')}$. An examination of Eq. (2.7) shows that $\alpha' \cong \gamma', \beta' \cong \delta'$, so they give atom-equivalent left transpositions. □

Example 2.61. Note that, unlike for 1-atoms, adjacent 2-atoms that are left and right transposable need not have the same left and right transposition. Let $\mathbf{x} : a \to a, \mathbf{y} : b \to b$, and $\mathbf{z} : c \to c$ be 1-primitives. Let $\mathbf{s}$ and $\mathbf{t}$ be 2-primitives with $\mathrm{s}(\mathbf{s}) = a \otimes \mathbf{y}$, $\mathrm{t}(\mathbf{s}) = \mathbf{x} \otimes b$ and $\mathrm{s}(\mathbf{t}) = \mathrm{t}(\mathbf{t}) = b \otimes \mathbf{z}$. Consider the adjacent 2-atoms $\eta = (\mathbf{s} \otimes c) \star (a \otimes b \otimes \mathbf{z})$ and $\epsilon = (\mathbf{x} \otimes b \otimes c) \star (a \otimes \mathbf{t})$ where for clarity, we've omitted notation for identity molecule components.

Note that $\eta \cong \eta' = (a \otimes b \otimes \mathbf{z}) \star (\mathbf{s} \otimes c)$ and $\epsilon \cong \epsilon' = (a \otimes \mathbf{t}) \star (\mathbf{x} \otimes b \otimes c)$; the 1-primitive in each atom "misses" the 2-primitive, thus can be transposed past its collapse. We can left transpose $\eta \circ \epsilon$ by choosing $\alpha = \mathbf{s} \otimes c$ and $\beta = a \otimes \mathbf{t}$ to get $\eta_l \circ \epsilon_l$ where $\eta_l = (a \otimes \mathbf{y} \otimes c) \star (a \otimes \mathbf{t})$ and $\epsilon_l = \eta$. Similarly we can right transpose η' and ϵ' (and thus η and ϵ) with the reversed choice of $\alpha' = a \otimes \mathbf{t}$ and $\beta = \mathbf{s} \otimes c$ to get $\eta_r \circ \epsilon_r$ where $\eta_r = a \otimes \mathbf{t} \star (a \otimes \mathbf{y} \otimes c)$ and $\epsilon_r = (a \otimes b \otimes \mathbf{z}) \star (\mathbf{s} \otimes c)$. But in general, $[\eta_l \circ \epsilon_l] \neq [\eta_r \circ \epsilon_r]$ as can be seen by examining their collapses; the source of $\mathbf{s}$ can not be transposed in the collapse in the same way as the target could, and ends up on different sides of η_l and η_r.

As in the 1-dimensional case, we can now restate the main result of [Mak05] for dimension 2:

Theorem 2.62. *Two non-identity 2-molecules f, g are equivalent if and only if $[f]$ can be transformed into $[g]$ by a finite sequence of left and/or right transpositions.*

Definition 2.63. Define ***i*-left transposition** of 2-molecules to be the partial operation on AE-2-molecules of left transposing the atoms at index i and $i+1$; this is defined if and only if the AE-2-molecule is of length at least $i+2$, and the designated indices are left transposable. Denote this operation by τ_i^{ℓ}. Define the ***i*-right transposition** similarly and denote it τ_i^r.

Definition 2.64. Let f, g be 2-molecules. We will call a finite list $(\tau_{k_i}^{d_i})_{i\in I}$ with $d_i \in \{\ell, r\}$ a **witness** for equivalence between f and g if the composite operation of applying them in order from left to right maps $[f]$ to $[g]$. We will write $\mu : f \rightsquigarrow g$ to indicate μ is a witness mapping f to g, and $\mu^{-1} : g \rightsquigarrow f$ will be the corresponding witness mapping g to f. We will similarly say that μ is a witness for equivalence between $[f]$ and $[g]$.

Note that, unlike for 1-molecules, two witnesses $\mu : f \rightsquigarrow g$ and $\nu : f \rightsquigarrow h$ can have the same underlying permutation while $g \ncong h$; see Example 2.61. We expect that the underlying braid may be a sufficient invariant for this purpose, but don't explore this any further.

2.13 Recognition principles for 2-molecule decompositions

Theorem 2.65. *A 2-molecule f in a non-degenerate strict monoidal n-computad is equivalent to one of the form $g \star h$ for two 2-molecules $g, h \neq 1_{1_a}$ iff at least one of the following holds:*

(i) $f \cong 1_{c\star d}$ for non-identity 1-molecules c, d; in this case $g = 1_c$, $h = 1_d$.

(ii) There exists a 1-molecule a s.t. every 2-atom of f is equivalent to one which has a as a prefix; in this case $g = 1_a$, and h is the 2-molecule coming from removing the prefix on each 2-atom.

(iii) There exists a 1-molecule b s.t. every 2-atom of f is equivalent to one which has b

as a suffix; in this case $h = 1_b$*, and* g *is the 2-molecule coming from removing the suffix on each 2-atom.*

(iv) f *is equivalent to a 2-molecule* m *such that there exists* $1 \leq p \leq \mathrm{length}(f) - 1$ *with* $\Delta_p \cdots \Delta_{\mathrm{length}(f)-1}$ *a valid transposition sequence on* m*, where* $\Delta_i = (\tau^{\ell}_{i-1}, \ldots, \tau^{\ell}_{i-p})$*.*

Proof. $(\Rightarrow)$ These can be seen directly from the definition of $\star$.

$(\Leftarrow)$ The implication is clear for (i) through (iii). For (iv), we will mimic the proof of Theorem 2.44. Let $q = \mathrm{length}(f) - p$. Take a choice of m such that the first p atoms in m are in righthand form and the rest of the atoms in m are in lefthand form, and let $m_i = l_1 \star (a_i \otimes \mathbf{m_i} \otimes b_i) \star r_i$. We will prove by induction on q that there exist 2-molecules g, h and 1-molecule w such that $[m] \rightsquigarrow [(1_{\mathrm{s}(g)} \star 1_w \star h) \circ (g \star 1_w \star 1_{\mathrm{t}(h)})]$ via $\Delta_p \cdots \Delta_{p+q-1}$, g does not have any shared 1-molecule suffix under atom equivalence (as in case (iii)), and h does not have any shared 1-molecule prefix under atom equivalence (as in case (ii)).

Assume $q = 1$. We know $(\tau^{\ell}_{p-1}, \ldots, \tau^{\ell}_0)$ is a valid transposition sequence on m. Since τ^{ℓ}_{p-1} is a valid operation on m, there exists x_{p-1} such that $r_{p-1} \cong x_{p-1} \star (a_p \otimes \mathrm{s}(\mathbf{m_p}) \otimes b_p) \star r_p$. Let $n = \tau^{\ell}_{p-1}(m)$; then $n_{p-1} \cong l_{p-1} \star (a_{p-1} \otimes \mathrm{s}(\mathbf{m_{p-1}}) \otimes b_{p-1}) \star x_{p-1} \star (a_p \otimes \mathbf{m_p} \otimes b_p) \star r_p$. If $p > 1$, then τ^{ℓ}_{p-2} is a valid operation on n, so there exists x_{p-2} such that $r_{p-2} \cong x_{p-2} \star (a_p \otimes \mathrm{s}(\mathbf{m_p}) \otimes b_p) \star r_p$. Proceeding similarly, there exists x_i such that $r_i \cong x_i \star (a_p \otimes \mathrm{s}(\mathbf{m_p}) \otimes b_p) \star r_p$ for each $0 \leq i \leq p - 1$. Hence by cancellation $l_i \star (a_i \otimes \mathrm{t}(\mathbf{m_i}) \otimes b_i) \star x_i \cong l_{i+1} \star (a_{i+1} \otimes \mathrm{s}(\mathbf{m_{i+1}}) \otimes b_{i+1}) \star x_{i+1}$ for $0 \leq i \leq p - 2$. Let $\widetilde{g}_i = l_i \star (a_i \otimes \mathbf{m_i} \otimes b_i) \star x_i$, and let $\widetilde{g} = \widetilde{g}_0 \star \cdots \star \widetilde{g}_{p-1}$. Note that the atoms of $\widetilde{g}$ are still in righthand form; let w be a maximal shared 1-molecule suffix of $\widetilde{g}$, such that $[\widetilde{g}] = [g \star 1_w]$. Let $h = (a_p \otimes \mathbf{m_p} \otimes b_p) \star r_p$. Then $[m] = [g \star 1_w \star h]$; from the definition, we can see that $\Delta_p : [m] \rightsquigarrow [(1_{\mathrm{s}(g)} \star 1_w \star h) \circ (g \star 1_w \star 1_{\mathrm{t}(h)})]$ as desired.

Now assume $q > 1$. By the induction hypothesis, we have $[m_0 \circ \cdots \circ m_{p+q-2}] \rightsquigarrow [(1_{\mathrm{s}(g)} \star 1_w \star h) \circ (g \star 1_w \star 1_{\mathrm{t}(h)})]$ via $\Delta_p \cdots \Delta_{p+q-2}$. Let $g \star 1_w \star 1_{\mathrm{t}(h)} = k_0 \circ \cdots \circ k_{p-1}$ where $k_i = l_i \star (a_i \otimes \mathbf{m_i} \otimes b_i) \star c_i \star w \star \mathrm{t}(h)$ for some c_i. Now consider $k_0 \circ \cdots \circ k_{p-1} \circ m_{p+q-1}$; since Δ_{p+q-1} is a valid operation on it, we can reason as in the $q = 1$ case to see that for all i, there exist

x_i such that $c_i \star w \star \mathrm{t}(h) \cong x_i \star (a_{p+q-1} \otimes \mathrm{s}\big(\mathbf{m_{p+q-1}}\big) \otimes b_{p+q-1}) \star r_{p+q-1}$. By Corollary 2.53, there exist minimal α, β such that $\alpha \star w \star \mathrm{t}(h) \cong \beta \star (a_{p+q-1} \otimes \mathrm{s}\big(\mathbf{m_{p+q-1}}\big) \otimes b_{p+q-1}) \star r_{p+q-1}$, $c_i \cong c_i' \star \alpha$, and $x_i \cong x_i' \star \beta$. Since α would be a shared 1-molecule suffix of g, it must be trivial; thus $w \star \mathrm{t}(h) \cong \beta \star (a_{p+q-1} \otimes \mathrm{s}\big(\mathbf{m_{p+q-1}}\big) \otimes b_{p+q-1}) \star r_{p+q-1}$.

Let $\mu : w \star \mathrm{t}(h) \rightsquigarrow \beta \star (a_{p+q-1} \otimes \mathrm{s}\big(\mathbf{m_{p+q-1}}\big) \otimes b_{p+q-1}) \star r_{p+q-1}$ be a witness. Let w have tag list C, and let C' be the sublist of tags landing in the indices of β under μ. Then we can choose a rearrangement $w \rightsquigarrow \widetilde{w}$ such that C' is a prefix of $\widetilde{w}$'s induced tag list, since any tag to the left of C' in $w \star \mathrm{t}(h)$ must be interchangeable with C' since it ends up to the right in $\beta \star (a_{p+q-1} \otimes \mathrm{s}\big(\mathbf{m_{p+q-1}}\big) \otimes b_{p+q-1}) \star r_{p+q-1}$. Similarly, there exists a rearrangement $\beta \rightsquigarrow \widetilde{\beta}$ so that $\widetilde{\beta}$ has prefix tag list C'. Thus $\widetilde{w} \star \mathrm{t}(h) \rightsquigarrow \widetilde{\beta} \star (a_{p+q-1} \otimes \mathrm{s}\big(\mathbf{m_{p+q-1}}\big) \otimes b_{p+q-1}) \star r_{p+q-1}$ has underlying permutation which acts as the identity on C', so $\widetilde{w} = \gamma \star \widetilde{w}_0$ and $\widetilde{\beta} = \gamma \star \widetilde{\beta}_0$. Then we can choose $g' = g$, $w' = \gamma$, and $h' = h \circ (\widetilde{\beta}_0 \star (a_{p+q-1} \otimes \mathrm{s}\big(\mathbf{m_{p+q-1}}\big) \otimes b_{p+q-1}) \star r_{p+q-1})$ to finish the induction.

We may turn a decomposition $g \star 1_w \star h$ into a binary decomposition by taking any partition $w \cong w' \star w''$, so that $f \cong (g \star 1_{w'}) \star (1_{w''} \star h)$. □

Before proceeding to the second main theorem, we need some preliminaries. The following result shows that tensor decompositions of 1-molecules are confluent; no matter how we decompose, we end up with the same list of components.

Lemma 2.66. *Let f be a 1-molecule such that $f \cong \otimes_{0 \le i \le m} g_i \cong \otimes_{0 \le j \le n} h_j$ and each $g_i, h_i \neq 1_\emptyset$. If each g_i, h_i is minimal, i.e. can not be written in the form $u \otimes v$ for $u, v \neq 1_\emptyset$, then $m = n$ and $g_i \cong h_i$ for all i.*

Proof. If $f = 1_{a_0 \otimes \cdots \otimes a_\ell}$, then by minimality and length we must have $g_i = h_i = 1_{a_i}$.

Assume f is not an identity. If $g_0 = 1_a$ for some 0-molecule a, then by minimality a is a single 0-primitive. Thus every 1-atom in $\otimes_i g_i$ has a as a prefix. Since transposition does not affect shared prefixes, so does every 1-atom in $\otimes_j h_j$. If $h_0 = 1_b$ we must have $a = b$, since a and b would both be shared single 0-primitive prefixes of f. If h_0 is not an identity, then every 1-atom in $h_0 \otimes (\otimes_{1\le j\le n} 1_{s(h_j)})$ has a as a shared prefix, and so every 1-atom in h_0 does as well. So by Theorem 2.44 $h_0 = 1_a \otimes k$ for some k, a contradiction to minimality. Therefore, g_0 must be a non-identity 1-molecule. Similarly, h_0 must also be a non-identity 1-molecule. Both must have no non-trivial 0-molecule prefix or suffix by minimality.

Choose witnesses $\mu : f \rightsquigarrow \otimes_i g_i$ and $\nu : f \rightsquigarrow \otimes_j h_j$. Tag f with tag list Z, giving the other 1-molecules the induced tag lists. Let $p = g_0 \otimes (\otimes_{1\le i\le m} 1_{s(g_i)})$ and $q = h_0 \otimes (\otimes_{1\le j\le n} 1_{s(h_j)})$ be the corresponding leading 1-molecules. Let A be the tag list thus associated to p and C the tag list associated to q. Note that $\otimes_i g_i$ then has tag list AB for some B and $\otimes_j h_j$ has tag list CD for some D.

Let $S(T)$ denote the underlying set of a given tag list T. By the form of $\otimes$, $S(A)$ and $S(C)$ are both minimal non-empty tag sets that are left interchangeable in f with the remainder of $S(Z)$; that is, left interchangeable in f with $S(B) = S(Z) \setminus S(A)$ and $S(D) = S(Z) \setminus S(C)$ respectively. To see this, note that if there were some smaller non-empty tag set $X \subset S(A)$ left interchangeable with $S(Z) \setminus X$ in f, then it would also be left interchangeable with $S(A) \setminus X$ in the submolecule p, and therefore in the 1-molecule g_0 given the same tag list. Thus, g_0 would be decomposable by Theorem 2.44, a contradiction to minimality. The same reasoning applies to $S(C)$.

Assume $S(A) \neq S(C)$, and let $Y = S(A) \cap S(C)$. Then Y is interchangeable with $S(Z) \setminus Y$ by Lemma 2.42, since it is left interchangeable with both $S(Z) \setminus S(A)$ and $S(Z) \setminus S(C)$. Furthermore, it is left interchangeable with $S(Z) \setminus Y$ from the definition. Thus it must be empty, since $S(A)$ and $S(C)$ were minimal. Therefore $S(A)$ and $S(C)$ are disjoint, so A and C are non-empty tag lists that are both left interchangeable with one another. Thus whenever we have a 1-molecule f' tag equivalent to f in which $a \prec c$ are adjacent tags in A and C respectively, then the associated atoms are left transposable. But the 1-molecule f'' coming from their left transposition has $c \prec a$, so the associated atoms in f'' are also left transposable. Since left and right transposition are inverses on adjacent pairs, this implies that the atoms associated to a and c are right transposable as well in f'. By Lemma 2.45, this is a contradiction to non-degeneracy. Therefore, we must have had $S(A) = S(C)$.

We have an equivalence $\nu \circ \mu^{-1} : \otimes_i g_i \rightsquigarrow \otimes_j h_j$ which has underlying permutation mapping A onto C; thus by Corollary 2.36, $p \cong q$. By minimality, the shared non-trivial suffix of this 1-molecule is then $\mathrm{s}(g_1) \otimes \cdots \otimes \mathrm{s}(g_m) = \mathrm{s}(h_1) \otimes \cdots \otimes \mathrm{s}(h_n)$; cancelling this, we see that $g_0 \cong h_0$ as desired.

If $m, n \geq 1$, by cancellation, we can proceed inductively on the remainder $\mathrm{t}(g_0) \otimes (\otimes_{1 \leq i \leq m} g_i)$ and $\mathrm{t}(h_0) \otimes (\otimes_{1 \leq j \leq n} h_j)$ by removing the shared prefix $\mathrm{t}(g_0) = \mathrm{t}(h_0)$. So assume without loss of generality that $m = 0$. If $n = 0$, we're done. If $n > 0$, then by cancellation, $1_{\mathrm{t}(g_0)} \cong 1_{\mathrm{t}(g_0)} \otimes (\otimes_{1 \leq i \leq n} h_i)$, so each h_i is an identity and each $\mathrm{s}(h_i)$ must be $\emptyset$, a contradiction to non-degeneracy. □

Definition 2.67. A 1-molecule x is said to be a **tensor-prefix** of a 2-atom α if $\alpha \cong 1_x \otimes \alpha'$ and $x \neq 1_\emptyset$. Note that the property of having x as a tensor-prefix is invariant under 2-atom equivalence. **Tensor-suffix** is defined similarly.

Lemma 2.68. *Let α, β, γ be 2-atoms and x, y be 1-molecules such that $\alpha \cong 1_x \otimes \beta$ and $\alpha \cong 1_y \otimes \gamma$, where both β and γ have no tensor-prefix. Then $x \cong y$ and $\beta \cong \gamma$. The same result holds for tensor-suffixes.*

Proof. We prove the result only for tensor-prefixes, since the tensor-suffix case is analogous. By definition of atom-equivalence, we have equivalence of collapses $\overline{\alpha} \cong x \otimes \overline{\beta} \cong y \otimes \overline{\gamma}$. By iterative decomposition, we can write $x \cong \otimes_{0 \leq i \leq m} x_i$, $\overline{\beta} \cong \otimes_{0 \leq i \leq n} z_i$, $y \cong \otimes_{0 \leq i \leq p} y_i$, and $\overline{\gamma} \cong \otimes_{0 \leq i \leq q} w_i$ where each of x_i, y_i, z_i, and w_i has no non-trivial $\otimes$-decomposition.

Note that if there were a non-trivial decomposition $\overline{\beta} \cong u \otimes v$ such that v contained the collapse primitive, then β would have have a tensor-prefix; thus we must have that the collapse primitive is in z_0, and reasoning similarly in w_0. By Lemma 2.66, we have that $m + n = p + q$ and the summands of $(\otimes_{0 \leq i \leq m} x_i) \otimes (\otimes_{0 \leq i \leq n} z_i) \cong (\otimes_{0 \leq i \leq p} y_i) \otimes (\otimes_{0 \leq i \leq q} w_i)$

are pairwise equivalent. Since the collapse primitive occurs in only one summand on each side, we must have $z_0 \cong w_0$ and $z_0 \ncong y_i$ for all i, $z_0 \ncong w_j$ for $j > 0$. Thus $m = p, n = q$, so $x \cong \otimes_{0 \le i \le m} x_i \cong \otimes_{0 \le i \le p} y_i = y$ and $\overline{\beta} \cong \otimes_{0 \le i \le n} z_i \cong \otimes_{0 \le i \le q} w_i \cong \overline{\gamma}$. Thus $x \cong y$ and $\beta \cong \gamma$ as desired. □

Lemma 2.69. *Let f, g, h be 2-molecules and x, y be 1-molecules such that $[f] = [1_x \otimes g] = [1_y \otimes h]$ and neither g nor h is atom-equivalent to $1_z \otimes \ell$ for any ℓ and $z \neq 1_{\emptyset}$. Then $x \cong y$, and $[g] = [h]$. The same result holds for tensor-suffixes.*

Proof. Since $[1_x \otimes g] = [1_y \otimes h]$, for $0 \le i < \mathrm{length}(f)$ we have $1_x \otimes g_i \cong 1_y \otimes h_i$. Choose maximal tensor-prefixes of g_i and h_i so that $g_i \cong 1_{u_i} \otimes g_i'$ and $h_i \cong 1_{v_i} \otimes h_i'$. By Lemma 2.68, we have $g_i' \cong h_i'$ and $x \otimes u_i \cong y \otimes v_i$ for each i.

Iteratively decompose to get $x = \otimes_{0 \le j \le m} x^j$, $y = \otimes_{0 \le j \le n} y^j$, $u_i \cong \otimes_{0 \le j \le p_i} u_i^j$, and $v_i \cong \otimes_{0 \le j \le q_i} v_i^j$ with each x^j, y^j, u_i^j, and v_i^j indecomposable. By Lemma 2.66, we have $m + p_i = n + q_i$ and the corresponding components are equivalent.

Assume $m \neq n$, and without loss of generality consider $m < n$. Then for all i, $y \cong x \otimes (\otimes_{0 \le j < n-m} u_i^j)$; thus by cancellation, $\otimes_{0 \le j < n-m} u_i^j$ is independent of i up to equivalence. Let u be this value for $i = 0$. Then u is a shared tensor prefix of g, a contradiction. Thus we must have $m = n$. Therefore, $x \cong y$ and $u_i^j \cong v_i^j$. Hence $g_i \cong h_i$ as well, so $[g] = [h]$. □

Definition 2.70. Let $\alpha = l_\alpha \star (a_\alpha \otimes \mathbf{p}_\alpha \otimes b_\alpha) \star r_\alpha$ and $\beta = l_\beta \star (a_\beta \otimes \mathbf{p}_\beta \otimes b_\beta) \star r_\beta$ be adjacent left transposable 2-atoms such that α is in righthand form, β is in lefthand form. Then by Lemma 2.59 there exists x such that $\mathrm{t}(\alpha) \cong \mathrm{s}(\beta) \cong l_\alpha \star (a_\alpha \otimes \mathrm{t}(\mathbf{p}_\alpha) \otimes b_\alpha) \star x \star (a_\beta \otimes \mathrm{s}(\mathbf{p}_\beta) \otimes b_\beta) \star r_\beta$. Let $\overline{\mathbf{p}_\alpha}$ and $\overline{\mathbf{p}_\beta}$ be distinct 1-primitives with the same source/target as the collapses of $\mathbf{p}_\alpha$ and $\mathbf{p}_\beta$ respectively. Then the 1-molecule $l_\alpha \star (a_\alpha \otimes \overline{\mathbf{p}_\alpha} \otimes b_\alpha) \star x \star (a_\beta \otimes \overline{\mathbf{p}_\beta} \otimes b_\beta) \star r_\beta$ will be known as a **shared collapse** of α, β. Shared collapses are unique up to equivalence, since all such x are equivalent.

Note that even in the case $\mathbf{p}_\alpha = \mathbf{p}_\beta$, we choose distinct shared collapse primitives such that $\overline{\mathbf{p}_\alpha} \neq \overline{\mathbf{p}_\beta}$. Also note that the above definition makes use only of left transposability;

we do not define a shared collapse for right transposable 2-atoms. We will only make use of shared collapses in situations where the left and right transposition both exist and are equal, as discussed below; thus we could have included this as part of the definition if we desired.

Definition 2.71. Let α, β be two adjacent left transposable 2-atoms with primitives p_α and p_β. Say that α, β are **left tensor transposable** if their shared collapse is equivalent to some $g \otimes h$ such that g contains $\overline{\mathbf{p}_\alpha}$ and h contains $\overline{\mathbf{p}_\beta}$, where these are the shared collapse primitives. They are **right tensor transposable** if the same holds, except g contains $\overline{\mathbf{p}_\beta}$ and h contains $\overline{\mathbf{p}_\alpha}$.

Lemma 2.72. *If α, β are left tensor transposable, then there exists g', h' such that $[\alpha \circ \beta] = [g' \otimes h']$. Thus they are also right transposable, and their left and right transpositions agree.*

Proof. By left tensor transposability, α and β have a shared collapse f of the form $l_\alpha \star (a_\alpha \otimes \overline{\mathbf{p}_\alpha} \otimes b_\alpha) \star x \star (a_\beta \otimes \overline{\mathbf{p}_\beta} \otimes b_\beta) \star r_\beta$ which is equivalent to some $g \otimes h$ such that $\overline{\mathbf{p}_\alpha}$ is in g and $\overline{\mathbf{p}_\beta}$ is in h.

Define a 2-atom g' from g by substituting $\mathbf{p}_\alpha$ for $\overline{\mathbf{p}_\alpha}$, and define h' by substituing $\mathbf{p}_\beta$ for $\overline{\mathbf{p}_\beta}$, so that $g = \overline{g'}$ and $h = \overline{h'}$. Then we can extend a witness $\mu : f \rightsquigarrow g \otimes h$ to proofs that $\alpha \cong g' \otimes \mathrm{s}(h')$ and $\beta \cong \mathrm{t}(g') \otimes h'$ – to see this, note that a proof $\alpha \cong g' \otimes \mathrm{s}(h')$ as 2-atoms comes from a witness $\overline{\alpha} \rightsquigarrow g \otimes \mathrm{s}(h')$. We can construct such a witness by replacing $\overline{\mathbf{p}_\beta}$ with $\mathrm{s}(\mathbf{p}_\beta)$ in f and then replacing every transposition in μ involving $\overline{\mathbf{p}_\beta}$ by a transposition sequence which block transposes $\mathrm{s}(\mathbf{p}_\beta)$. The same construction works to show $\beta \cong \mathrm{t}(g') \otimes h'$.

Since $(g' \otimes \mathrm{s}(h')) \circ (\mathrm{t}(g') \otimes h') = g' \otimes h'$, we have $[\alpha \circ \beta] = [(g' \otimes \mathrm{s}(h')) \circ (\mathrm{t}(g') \otimes h')] = [g' \otimes h']$ as desired. Their left and right transpositions agree, since this is true in general for tensors of 2-atoms. □

Theorem 2.73. *A 2-molecule f in a non-degenerate strict monoidal n-computad is equivalent to one of the form $g \otimes h$ for two 2-molecules $g, h \neq 1_{1_\emptyset}$ iff at least one of the following holds:*

(i) $f \cong 1_{c \otimes d}$ for 1-molecules $c, d \neq 1_\emptyset$; in this case $g = 1_c$, $h = 1_d$.

(ii) There exists a 1-molecule a s.t. every 2-atom of f has a as as tensor-prefix; in this case $g = 1_a$, and h is the 2-molecule coming from removing the tensor-prefix on each 2-atom.

(iii) There exists a 1-molecule b s.t. every 2-atom of f has b as a tensor-suffix; in this case $h = 1_b$, and g is the 2-molecule coming from removing the tensor-suffix on each 2-atom.

(iv) f is equivalent to a 2-molecule m such that there exists $1 \leq p \leq \mathrm{length}(f) - 1$ with $\Delta_p \cdots \Delta_{\mathrm{length}(f)-1}$ a valid transposition sequence on m, where $\Delta_i = (\tau^\ell_{i-1}, \dots, \tau^\ell_{i-p})$, and each transposition is of left tensor transposable pairs.

Proof. (⇒) These can be seen directly from the definition of $\otimes$.

(⇐) The implication is clear for (i) through (iii).

Let $q = \mathrm{length}(f) - p$. We will prove by induction on q that there exist 2-molecules g, h and a 1-molecule w such that $[m] \rightsquigarrow [(1_{\mathsf{s}(g)} \otimes 1_w \otimes h) \circ (g \otimes 1_w \otimes 1_{\mathsf{t}(h)})]$ via $\Delta_p \cdots \Delta_{p+q-1}$, g does not have any shared 1-molecule tensor-suffix (as in case (iii)), and h does not have any shared 1-molecule tensor-prefix (as in case (ii)).

Assume $q = 1$. We know $(\tau^\ell_{p-1}, \dots, \tau^\ell_0)$ is a valid transposition sequence on m via left tensor transposable pairs. Thus m_{p-1} and m_p are left tensor transposable, so by Lemma 2.72 we have $[m_{p-1} \circ m_p] = [u'_{p-1} \otimes v'_{p-1}]$ for some 2-atoms u'_{p-1} and v'_{p-1}. By Lemma 2.68, we can choose a maximal tensor-suffix c_{p-1} of u'_{p-1} and a maximal tensor-prefix d_{p-1} of v'_{p-1} so that $[u'_{p-1}] = [u_{p-1} \otimes 1_{c_{p-1}}]$ and $[v'_{p-1}] = [1_{d_{p-1}} \otimes v_{p-1}]$. The left or right transposition of $[m_{p-1} \circ m_p]$ is then $[(1_{\mathsf{s}(u_{p-1}) \otimes c_{p-1} \otimes d_{p-1}} \otimes v_{p-1}) \circ (u_{p-1} \otimes 1_{c_{p-1} \otimes d_{p-1} \otimes \mathsf{t}(v_{p-1})})]$. We thus know that m_{p-2} is left tensor transposable with $1_{\mathrm{s}(u_{p-1})\otimes c_{p-1}\otimes d_{p-1}} \otimes v_{p-1}$, so we have $[m_{p-2}\circ(1_{\mathrm{s}(u_{p-1})\otimes c_{p-1}\otimes d_{p-1}} \otimes v_{p-1})] = [u'_{p-2} \otimes v'_{p-2}]$. Again, choose maximal tensor-suffixes and tensor-prefixes so that $[u'_{p-2}] = [u_{p-2} \otimes 1_{c_{p-2}}]$ and $[v'_{p-2}] = [1_{d_{p-2}} \otimes v_{p-2}]$. Continuing, we can define u_{p-i}, c_{p-i}, d_{p-i}, and v_{p-i} for $1 \leq i \leq p$. Since $[m_{p-i-1}\circ(1_{\mathrm{s}(u_{p-i})\otimes c_{p-i}\otimes d_{p-i}} \otimes v_{p-i})] = [u_{p-i-1} \otimes 1_{c_{p-i-1}} \otimes 1_{d_{p-i-1}} \otimes v_{p-i-1}]$, their righthand 2-atoms are equivalent; thus $1_{\mathrm{s}(u_{p-i})\otimes c_{p-i}\otimes d_{p-i}} \otimes v_{p-i} \cong 1_{\mathrm{t}(u_{p-i-1})\otimes c_{p-i-1}\otimes d_{p-i-1}} \otimes v_{p-i-1}$. By Lemma 2.68, $v_{p-i} \cong v_{p-i-1}$ and $\mathrm{s}\left(u_{p-i}\right) \otimes c_{p-i} \otimes d_{p-i} \cong \mathrm{t}\left(u_{p-i-1}\right) \otimes c_{p-i-1} \otimes d_{p-i-1}$. Let $g'_i = u_i \otimes 1_{c_i\otimes d_i}$ and $g' = \circ_i g'_i$. Let w be a maximal shared tensor-suffix of g', so that $[g'] = [g \otimes 1_w]$. Let $h = v_0$. Then we can verify that $[m] = [g \otimes 1_w \otimes h]$, so that Δ_p maps $[m]$ to $[(1_{\mathrm{s}(g)} \otimes 1_w \otimes h)\circ(g \otimes 1_w \otimes 1_{\mathrm{t}(h)})]$ as desired.

Now assume $q > 1$. By the induction hypothesis, we have $[m_0\circ \cdots \circ m_{p+q-2}] \rightsquigarrow [(1_{\mathrm{s}(g)}\otimes 1_w \otimes h)\circ(g \otimes 1_w \otimes 1_{\mathrm{t}(h)})]$ via $\Delta_p \cdots \Delta_{p+q-2}$. Let $g \otimes 1_w \otimes 1_{\mathrm{t}(h)} = k_0\circ \cdots \circ k_{p-1}$ and consider $k_0\circ \cdots \circ k_{p-1}\circ m_{p+q-1}$; since Δ_{p+q-1} is a valid operation on it, we can reason as in the $q = 1$ case to see that $[k_0\circ \cdots \circ k_{p-1}\circ m_{p+q-1}] = [g' \otimes 1_{w'} \otimes h']$ for some tensor-minimal g', h' with $\mathrm{length}(g') = p$, $\mathrm{length}(h') = 1$. Thus $[g \otimes 1_w \otimes 1_{\mathrm{t}(h)}] = [g' \otimes 1_{w'} \otimes 1_{\mathrm{s}(h')}]$. By minimality and Lemma 2.69, we have $[g] = [g']$ and $w \otimes \mathrm{t}(h) \cong w' \otimes \mathrm{s}\left(h'\right)$. Applying Δ_{p+q-1}, we see that $[m] \rightsquigarrow [(1_{\mathrm{s}(g)} \otimes 1_w \otimes h)\circ(1_{\mathrm{s}(g')} \otimes 1_{w'} \otimes h')\circ(g' \otimes 1_{w'} \otimes 1_{\mathrm{t}(h')})]$. Since $w \otimes \mathrm{t}(h) \cong w' \otimes \mathrm{s}\left(h'\right)$, let $\ell = (1_w \otimes h)\circ(1_{w'} \otimes h')$ and choose a maximal shared tensor-prefix of ℓ so that $\ell = 1_{w''} \otimes \ell'$. Then we have $[m] \rightsquigarrow [(1_{\mathrm{s}(g)} \otimes 1_{w''} \otimes \ell')\circ(g \otimes 1_{w''} \otimes 1_{\mathrm{t}(\ell')})]$, finishing the induction.

We may turn a decomposition $g \otimes 1_w \otimes h$ into a binary decomposition by taking any partition $w \cong w' \otimes w''$, so that $f \cong (g \otimes 1_{w'}) \otimes (1_{w''} \otimes h)$. □

2.14 Proof of Lemma 2.34

Lemma 2.74. *If $(\tau_{i+k}, \tau_{i+k-1}, \tau_{i+k})$ is a well defined transposition sequence on a 1-molecule, then so is $(\tau_{i+k-1}, \tau_{i+k}, \tau_{i+k-1})$, and they give the same result.*

Proof. It is enough to consider the sequences (τ_1, τ_0, τ_1) and (τ_0, τ_1, τ_0) on a 1-molecule h of length 3. Let $h_i = a_i \otimes \mathbf{h_i} \otimes b_i$. We will break up into eight cases, corresponding to whether each of the transpositions involved in (τ_1, τ_0, τ_1) is a left or right transposition; by symmetry, we can assume the first transposition is a left transposition, reducing to four cases.

Case (left, left, left): If all 3 transpositions (τ_1, τ_0, τ_1) are left transpositions, then since h_1 and h_2 are left transposable, there exists x_0 such that $b_1 = x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2$ and $a_2 = a_1 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0$. The result of this transposition is

$$\begin{array}{rl} a_0 \otimes \mathbf{h_0} \otimes b_0 & \star \\ a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathbf{h_2} \otimes b_2 & \star \\ a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 & \end{array}$$

The first two atoms are left transposable, so there exists x_1 such that $b_0 = x_1 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2$ and $a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 = a_0 \otimes \mathrm{t}(\mathbf{h_0}) \otimes x_1$. Transposing them, we have

$$\begin{array}{rl} a_0 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_1 \otimes \mathbf{h_2} \otimes b_2 & \star \\ a_0 \otimes \mathbf{h_0} \otimes x_1 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 & \star \\ a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 & \end{array}$$

The last two atoms are left transposable, so there exists x_2 such that $x_1 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 = x_2 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2$ and $a_1 = a_0 \otimes \mathrm{t}(\mathbf{h_0}) \otimes x_2$. By cancellation, $x_1 = x_2 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0$. Transposing them, we finally have

$$\begin{array}{rl} a_0 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_1 \otimes \mathbf{h_2} \otimes b_2 & \star \\ a_0 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_2 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 & \star \\ a_0 \otimes \mathbf{h_0} \otimes x_2 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 & \end{array}$$

Now, to see that τ_0 is valid on h, note that $b_0 = x_1 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2 = x_2 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes$

$\mathrm{s}(\mathbf{h_2}) \otimes b_2 = x_2 \otimes \mathrm{s}(\mathbf{h_1}) \otimes b_1$ and $a_1 = a_0 \otimes \mathrm{t}(\mathbf{h_0}) \otimes x_2$. Thus x_2 is a choice of x demonstrating left transposability. Transposing, we have

$$\begin{aligned} a_0 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_2 \otimes \mathbf{h_1} \otimes b_1 &\qquad \star \\ a_0 \otimes \mathbf{h_0} \otimes x_2 \otimes \mathrm{t}(\mathbf{h_1}) \otimes b_1 &\qquad \star \\ a_2 \otimes \mathbf{h_2} \otimes b_2 & \end{aligned}$$

To show that τ_1 is valid on this result, note that $a_2 = a_1 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0 = a_0 \otimes \mathrm{t}(\mathbf{h_0}) \otimes x_2 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0$ and $x_2 \otimes \mathrm{t}(\mathbf{h_1}) \otimes b_1 = x_2 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2$. Thus $x_2 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0$ is a choice of x demonstrating left transposability. Transposing, we have

$$\begin{aligned} a_0 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_2 \otimes \mathbf{h_1} \otimes b_1 &\qquad \star \\ a_0 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_2 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0 \otimes \mathbf{h_2} \otimes b_2 &\qquad \star \\ a_0 \otimes \mathbf{h_0} \otimes x_2 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 & \end{aligned}$$

To show that τ_0 is valid on this result, note that $b_1 = x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2$. Thus x_0 is a choice of x demonstrating left transposability. Transposing, we finally have

$$\begin{aligned} a_0 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_2 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathbf{h_2} \otimes b_2 &\qquad \star \\ a_0 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_2 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 &\qquad \star \\ a_0 \otimes \mathbf{h_0} \otimes x_2 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 & \end{aligned}$$

which is the same as of the result of the first transposition sequence.

Case (left,left,right): If the first two transpositions in (τ_1, τ_0, τ_1) are left transpositions and the last is right, then we may proceed similarly to get that there exists x_0, x_1 such that

$$\begin{aligned} b_1 &= x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2 \\ a_2 &= a_1 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0 \\ b_0 &= x_1 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2 \\ a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 &= a_0 \otimes \mathrm{t}(\mathbf{h_0}) \otimes x_1 \end{aligned}$$

and the result of the first two transpositions is

$$
\begin{aligned}
a_0 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes x_1 \otimes \mathbf{h_2} \otimes b_2 \qquad & \star \\
a_0 \otimes \mathbf{h_0} \otimes x_1 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes b_2 \qquad & \star \\
a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes b_2 \qquad &
\end{aligned}
$$

Since the last two atoms are right transposable, there exists x_2 such that $a_0 = a_1 \otimes \mathrm{s}\left(\mathbf{h_1}\right) \otimes x_2$ and $x_0 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes b_2 = x_2 \otimes \mathrm{t}\left(\mathbf{h_0}\right) \otimes x_1 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes b_2$. By cancellation, $x_0 = x_2 \otimes \mathrm{t}\left(\mathbf{h_0}\right) \otimes x_1$. Transposing, we get

$$
\begin{aligned}
a_0 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes x_1 \otimes \mathbf{h_2} \otimes b_2 \qquad & \star \\
a_1 \otimes \mathbf{h_1} \otimes x_2 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes x_1 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes b_2 \qquad & \star \\
a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_2 \otimes \mathbf{h_0} \otimes x_1 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes b_2 \qquad &
\end{aligned}
$$

Now, to see that τ_0 is valid on h, note that $a_0 = a_1 \otimes \mathrm{s}\left(\mathbf{h_1}\right) \otimes x_2$ and $b_1 = x_0 \otimes \mathrm{s}\left(\mathbf{h_2}\right) \otimes b_2 = x_2 \otimes \mathrm{t}\left(\mathbf{h_0}\right) \otimes x_1 \otimes \mathrm{s}\left(\mathbf{h_2}\right) \otimes b_2 = x_2 \otimes \mathrm{t}\left(\mathbf{h_0}\right) \otimes b_0$. Thus x_2 is a choice of x demonstrating right transposability. Transposing, we have

$$
\begin{aligned}
a_1 \otimes \mathbf{h_1} \otimes x_2 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes b_0 \qquad & \star \\
a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_2 \otimes \mathbf{h_0} \otimes b_0 \qquad & \star \\
a_2 \otimes h_2 \otimes b_2 \qquad &
\end{aligned}
$$

To show that τ_1 is valid on this result, note that $a_2 = a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_0 = a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_2 \otimes \mathrm{t}\left(\mathbf{h_0}\right) \otimes x_1$ and $b_0 = x_1 \otimes \mathrm{s}\left(\mathbf{h_2}\right) \otimes b_2$. Thus x_1 is a choice of x demonstrating left transposability. Transposing, we have

$$
\begin{aligned}
a_1 \otimes \mathbf{h_1} \otimes x_2 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes b_0 \qquad & \star \\
a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_2 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes x_1 \otimes \mathbf{h_2} \otimes b_2 \qquad & \star \\
a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_2 \otimes \mathbf{h_0} \otimes x_1 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes b_2 \qquad &
\end{aligned}
$$

To show that τ_0 is valid on this result, note that $b_0 = x_1 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2$, so $x_2 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_1$ is a choice of x demonstrating left transposability. Transposing, we have

$$\begin{aligned} a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_2 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_1 \otimes \mathbf{h_2} \otimes b_2 &\qquad \star \\ a_1 \otimes \mathbf{h_1} \otimes x_2 \otimes \mathrm{s}(\mathbf{h_0}) \otimes x_1 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 &\qquad \star \\ a_1 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_2 \otimes \mathbf{h_0} \otimes x_1 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 & \end{aligned}$$

which is the same as the result of the first transposition sequence.

Case (left,right,left): If (τ_1, τ_0, τ_1) are left, right, and left transpositions respectively when applied to h, then since h_1 and h_2 are left transposable, there exists x_0 such that $b_1 = x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2$ and $a_2 = a_1 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0$. The result of this transposition is

$$\begin{aligned} a_0 \otimes \mathbf{h_0} \otimes b_0 &\qquad \star \\ a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathbf{h_2} \otimes b_2 &\qquad \star \\ a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 & \end{aligned}$$

The first two atoms are right transposable, so there exists x_1 such that $a_0 = a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes x_1$ and $b_2 = x_1 \otimes \mathrm{t}(\mathbf{h_0}) \otimes b_0$. Transposing them and substituing for b_2, we have

$$\begin{aligned} a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathbf{h_2} \otimes x_1 \otimes \mathrm{s}(\mathbf{h_0}) \otimes b_0 &\qquad \star \\ a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes x_1 \otimes \mathbf{h_0} \otimes b_0 &\qquad \star \\ a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes x_1 \otimes \mathrm{t}(\mathbf{h_0}) \otimes b_0 & \end{aligned}$$

The last two atoms are manifestly right transposable, with x chosen to be $x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes x_1$. But by assumption, they are also left transposable. Examining the proof of Lemma 2.27, we see that $\mathrm{t}(\mathbf{h_0}) = \mathrm{s}(\mathbf{h_1}) = \emptyset$, a contradiction to non-degeneracy. Thus this case can't occur.

Case (left,right,right): If (τ_1, τ_0, τ_1) are left, right, and right transpositions respectively when applied to h, then since h_1 and h_2 are left transposable, there exists x_0 such that $b_1 = x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2$ and $a_2 = a_1 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0$. The result of this transposition is

$$\begin{aligned} a_0 \otimes \mathbf{h_0} \otimes b_0 \quad & \star \\ a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathbf{h_2} \otimes b_2 \quad & \star \\ a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes b_2 \quad & \end{aligned}$$

The first two atoms are right transposable, so there exists x_1 such that $a_0 = a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes x_1$ and $b_2 = x_1 \otimes \mathrm{t}(\mathbf{h_0}) \otimes b_0$. Transposing them and substituing for b_2, we have

$$\begin{aligned} a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathbf{h_2} \otimes x_1 \otimes \mathrm{s}(\mathbf{h_0}) \otimes b_0 \quad & \star \\ a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes x_1 \otimes \mathbf{h_0} \otimes b_0 \quad & \star \\ a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes x_1 \otimes \mathrm{t}(\mathbf{h_0}) \otimes b_0 \quad & \end{aligned}$$

The last two atoms are manifestly right transposable, with x chosen to be $x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes x_1$. Transposing them, we have

$$\begin{aligned} a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathbf{h_2} \otimes x_1 \otimes \mathrm{s}(\mathbf{h_0}) \otimes b_0 \quad & \star \\ a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes x_1 \otimes \mathrm{s}(\mathbf{h_0}) \otimes b_0 \quad & \star \\ a_1 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{t}(\mathbf{h_2}) \otimes x_1 \otimes \mathbf{h_0} \otimes b_0 \quad & \end{aligned}$$

Now, to see that τ_0 is valid on h, note that $a_0 = a_1 \otimes \mathrm{s}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes x_1$ and $b_1 = x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes b_2 = x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes x_1 \otimes \mathrm{t}(\mathbf{h_0}) \otimes b_0$. Thus $x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes x_1$ is a choice of x demonstrating right transposability. Transposing, we have

$$\begin{aligned} a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes x_1 \otimes \mathrm{s}(\mathbf{h_0}) \otimes b_0 \quad & \star \\ a_1 \otimes \mathrm{t}(\mathbf{h_1}) \otimes x_0 \otimes \mathrm{s}(\mathbf{h_2}) \otimes x_1 \otimes \mathbf{h_0} \otimes b_0 \quad & \star \\ a_2 \otimes h_2 \otimes b_2 \quad & \end{aligned}$$

To show that τ_1 is valid on this result, note that $a_2 = a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_0$ and $b_2 = x_1 \otimes \mathrm{t}\left(\mathbf{h_0}\right) \otimes b_0$. Thus x_1 is a choice of x demonstrating right transposability. Transposing, we have

$$\begin{aligned}
&a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{s}\left(\mathbf{h_2}\right) \otimes x_1 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes b_0 \qquad \star \\
&a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_0 \otimes \mathbf{h_2} \otimes x_1 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes b_0 \qquad \star \\
&a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_0 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes x_1 \otimes \mathbf{h_0} \otimes b_0
\end{aligned}$$

Now τ_0 is clearly valid on this result, with $x = x_0$ demonstrating left transposability. Transposing, we have

$$\begin{aligned}
&a_1 \otimes \mathrm{s}\left(\mathbf{h_1}\right) \otimes x_0 \otimes \mathbf{h_2} \otimes x_1 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes b_0 \qquad \star \\
&a_1 \otimes \mathbf{h_1} \otimes x_0 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes x_1 \otimes \mathrm{s}\left(\mathbf{h_0}\right) \otimes b_0 \qquad \star \\
&a_1 \otimes \mathrm{t}\left(\mathbf{h_1}\right) \otimes x_0 \otimes \mathrm{t}\left(\mathbf{h_2}\right) \otimes x_1 \otimes \mathbf{h_0} \otimes b_0
\end{aligned}$$

which is the same as the result of the first transposition sequence. □

Proof of Lemma 2.34. We will show that if the left hand side of any of the rewriting rules in Lemma 2.33 is defined on a given 1-molecule (when interpreted as applications of τ operations), then the right hand side is also defined and the two are equal as operations. This will imply that any transposition sequence can be reduced to a normal form without altering the underlying permutation or operation, and that two sequences with the same underlying permutation have the same normal form.

Going rule by rule, consider the potential actions on a 1-molecule g given by:

- $(\tau_i, \tau_i) \to () \quad (i \geq 0)$: The right hand side of this rule is always defined, and if g_i, g_{i+1} are left transposeable, then their transposition is right transposeable with the same choice of α, β. Transposing twice is the identity.

- $(\tau_i, \tau_j) \to (\tau_j, \tau_i) \quad (j \geq i + 2)$: Since $j \geq i + 2$, the two transpositions involve non-intersecting indices; thus applying one does not alter the atoms involved in the other. Thus, given that the operation on the left is defined on g, applying them in the other order is also defined and gives the same result.

- $(\tau_{i+k}, \tau_i, \tau_{i+1}, \ldots, \tau_{i+k-1}, \tau_{i+k}) \to (\tau_i, \tau_{i+1}, \ldots, \tau_{i+k-1}, \tau_{i+k}, \tau_{i+k-1}) \quad (i \geq 0, k \geq 1)$: Assume the left hand side is defined on g. Then given a tag list on g, every tag which would be involved in a transposition on the left is involved in the exact same transpositions in sequence on the right except for the ones at index i, $i+k$, and $i+k+1$. Let $h = (\tau_i, \ldots, \tau_{i+k-2})(g)$ and observe that $(\tau_{i+k}, \tau_{i+k-1}, \tau_{i+k})(h)$ is the same as $(\tau_{i+k}, \tau_i, \tau_{i+1}, \ldots, \tau_{i+k-1}, \tau_{i+k})(g)$. Now apply Lemma 2.74 to h.

□

Chapter 3

The `CatComputad` library

3.1 Summary of features and results

Before diving into the details on how to use `CatComputad` , we'll first summarize what it can do and what's been done with it. Primary functionality includes:

- Finding equivalence classes of 1-molecules, 2-atoms, and 2-molecules in a non-degenerate strict monoidal computad.
- Finding all decompositions of a k-molecule into ℓ-dimensional compositions, for $0 \leq \ell \leq k \leq 2$.
- Finding minimal size ASTs representing a k-cell for $0 \leq k \leq 2$.
- Finding all ways in which a primitive family's source matches a given k-cell, as outlined in Chapter 2.
- Finding all paths between two k-cells using a set of primitive families for $0 \leq k \leq 2$, assuming that there are finitely many. If $k = 2$ we can not equivalence check the 3 dimensional paths between them, so we may return the same path in multiple forms.

Using the library, we've been able to rederive (from their 1-dimensional boundaries) many of the 2-dimensional diagrams involved in the data and axioms of strict symmetric monoidal 2-categories and symmetric monoidal bihomomorphisms between them. We can also generate speculatively equivalent forms for such diagrams; such forms should give equivalent versions of definitions, although we do not attempt to prove such in any given case.

We have also been able to find a natural extension of one such family of diagrams, which should form a part of the definition of symmetric monoidal trihomomorphism between strict symmetric monoidal 3-categories. We'll exhibit this in Section 3.5.

3.2 Representing cells

CatComputad uses a variety of classes for cells and their equivalence classes. First, atomic primitives are represented by the ConstPrim classes:

```
a0 = ConstPrim0("a")
b0 = ConstPrim0("b")

f1 = ConstPrim1("f", a0, b0)
g1 = ConstPrim1("g", a0, b0)

u2 = ConstPrim2("u", f1, g1)
v2 = ConstPrim2("v", f1, g1)

p3 = ConstPrim3("p", u2, v2)
```

In dimension 0, the class Mol0 represents a molecule, which we can build in several ways.

```
y1 = prim0ToMol0(a0)

x1 = fMol0((a0, b0))
x2 = comp0(a0, b0)
x1 == x2
# >>> True
```

Note that we use a factory function fMol0 to directly get a Mol0 instance; this pattern is repeated throughout the codebase. We use instance identity checking for equality checking, which is the reason for this pattern. This will likely change in future versions. The comp0 function is part of a family compN, which for cells of any dimension gives their N-dimensional composition with the minimal appropriate output type.

In dimension 1, we have the Atom1, IdMol1, NonIdMol1, and EqMol1 classes. Most use cases will involve the EqMol1 class, which uses the others under the hood.

```
a, b, c, d = ConstPrim0("a"), ConstPrim0("b"), ConstPrim0("c"),
    ConstPrim0("d")
f, g = ConstPrim1("f", a, b), ConstPrim1("g", c, d)
emptyMol0 = fMol0(())

# Constructing an identity molecule.
idMol = fIdMol1(ensureMol0(a))
# Constructing the same equivalence class in two ways.
idMolEq = fEqMol1(idMol)
idMolEq2 = ensureEqMol1(a) # This automatically upcasts a to an identity
    molecule.
idMolEq == idMolEq2
# >>> True

# Constructing a pair of atoms in two different ways.
firstAtom = fAtom1(emptyMol0, f, ensureMol0(c))
secondAtom = fAtom1(ensureMol0(b), g, emptyMol0)
str(firstAtom)
# >>> 'f @ c'
str(secondAtom)
# >>> 'b @ g'
firstAtom2 = comp0(f, c)
secondAtom2 = comp0(b, g)
```

```
(firstAtom, secondAtom) == (firstAtom2, secondAtom2)
# >>> True

# Building a 1-molecule in two different ways.
mol = fNonIdMol1((firstAtom, secondAtom))
mol2 = comp1(firstAtom, secondAtom)
mol == mol2
# >>> True

# Finding the equivalence class of a 1-molecule.
eqMol = fEqMol1(mol)
print(lmap(str, eqMol.mol1s))
# >>> [’(f @ c) . (b @ g)’, ’(a @ g) . (f @ d)’]
```

As mentioned in Chapter 2, to ensure equivalence checking and decomposition works correctly, we can not make use of degenerate 1-primitives:

```
f = ConstPrim1("f", fMol0(()), fMol0(()))

# >>> Traceback (most recent call last):
# >>> ...
# >>> Exception: Degenerate 1-primitives disallowed.
```

In dimension 2, we have the additional trouble that 2-atom equivalence is different from equality. Thus we have different classes, `Atom2` and `EqAtom2`, representing an individual atom and its equivalence class. We do not bother to represent "raw" 2-molecules, as these are not involved in any of our use cases; instead we have 2-molecules up to atom-equivalence given by `AEIdMol2` and `AENonIdMol2`. Equivalence classes of 2-molecules use the class `EqAEMol2`.

```
a, b, c, d = ConstPrim0("a"), ConstPrim0("b"), ConstPrim0("c"),
    ConstPrim0("d")
f, g = ConstPrim1("f", a, b), ConstPrim1("g", a, b)
h, k = ConstPrim1("h", c, d), ConstPrim1("k", c, d)
u, v = ConstPrim2("u", f, g), ConstPrim2("v", h, k)

# Checking the equivalence class of a 2-atom.
atom = comp0(u, h)
eqAtom = fEqAtom2(atom)
# print(lmap(str, eqAtom.atom2s))
# >>> [’(u @ c) . (b @ h)’, ’(a @ h) . (u @ d)’]

# Checking the equivalence class of a 2-molecule.
# Note that & is used to represent 2-composition.
mol = comp0(u, v)
eqMol = ensureEqAEMol2(mol)
# print(lmap(str, eqMol.aeMol2s))
# >>> [’[(u @ c) . (b @ h)] & [(a @ v) . (g @ d)]’, ’[(a @ v) . (f @ d)] &
    [(u @ c) . (b @ k)]’]
```

3.3 Decompositions

Functions are provided for performing the various decompositions on equivalences classes as discussed in Chapter 2. Each function returns all non-trivial decompositions of the given type. For example, using the defined primitives above:

```
def displayDecomps(decomps):
   print(lmap(lambda x: (str(x[0]), str(x[1])), decomps))
```

```
td = tensorDecompEqAEMol2(ensureEqAEMol2(comp0(u,v)))
displayDecomps(td1)
# >>> [(’[[u]]’, ’[[v]]’)]

hd = horizontalDecompEqAEMol2(ensureEqAEMol2(comp0(u,v)))
displayDecomps(hd)
# >>> [(’[[(u @ c)]]’, ’[[(b @ v)]]’), (’[[(a @ v)]]’, ’[[(u @ d)]]’)]

vd = verticalDecompsEqAEMol2(ensureEqAEMol2(comp0(u,v)))
displayDecomps(vd)
# >>> [(’[[(u @ c) . (b @ h)]]’, ’[[(a @ v) . (g @ d)]]’), (’[[(a @ v) .
    (f @ d)]]’, ’[[(u @ c) . (b @ k)]]’)]
```

3.4 Filling example: unitors

We will illustrate usage of our filling algorithm by "rediscovering" some of the unitor diagrams occurring in the definition of a strict symmetric monoidal 2-category. Excellent illustrations in the more general case can be found in Appendix C of [SP09].

First, we have as base level data two unitors in dimension 1, $l : I \otimes A \to A$ and $r : A \otimes I \to A$. We create `PrimitiveFamily` objects to represent these, which take as arguments source and target ASTs. We create these ASTs by calling `minimalASTFromMol0` with a `Mol0` and a list of primitives we want to treat as variables in the final AST.

```
unit = ConstPrim0("unit")
a = ConstPrim0("a")
lSource = minimalASTFromMol0(comp0(unit, a), [a])
lTarget = minimalASTFromMol0(ensureMol0(a), [a])
rSource = minimalASTFromMol0(comp0(a, unit), [a])
rTarget = minimalASTFromMol0(ensureMol0(a), [a])
```

```
# Arguments in order:
#   name
#   dimension
#   functor count (explained later)
#   dimension of each parameter
#   source AST
#   target AST
u1_L = PrimitiveFamily("unitor1L", 1, 0, [0], lSource, lTarget)
u1_R = PrimitiveFamily("unitor1R", 1, 0, [0], rSource, rTarget)
```

Next, we have three 2-unitor families in dimension 2 constructed as follows: insert I into the list (A, B) via tensor at any of the three possible spots. There are two distinct paths mapping from the result to $A \otimes B$; these paths form the source and target of the desired cell.

```
b = ConstPrim0("b")
source1 = comp0s(unit, a, b)
target1 = comp0s(a, b)
paths = findPaths1(source1, target1, [u1_L, u1_R])
lmap(str, paths)
# >>> ['[(unitor1L((a @ b)))]', '[(unitor1L(a) @ b)]']

# NOTE: In practice, path-finding does not return paths in a deterministic
    order; thus we should should be careful to copy down the path if we
    want to reuse it between runs.
u2_0Source = minimalASTFromEqMol1(paths[0], [a, b])
u2_0Target = minimalASTFromEqMol1(paths[1], [a, b])
u2_0 = PrimitiveFamily("unitor2_0", 2, 0, [0, 0], u2_0Source, u2_0Target)
```

```
source2 = comp0s(a, unit, b)
target2 = comp0s(a, b)
paths = findPaths1(source2, target2, [u1_L, u1_R])
u2_1Source = minimalASTFromEqMol1(paths[0], [a, b])
u2_1Target = minimalASTFromEqMol1(paths[1], [a, b])
u2_1 = PrimitiveFamily("unitor2_2", 2, 0, [0, 0], u2_1Source, u2_1Target)

source3 = comp0s(a, b, unit)
target3 = comp0s(a, b)
paths = findPaths1(source3, target3, [u1_L, u1_R])
u2_2Source = minimalASTFromEqMol1(paths[0], [a, b])
u2_2Target = minimalASTFromEqMol1(paths[1], [a, b])
u2_2 = PrimitiveFamily("unitor2_2", 2, 0, [0, 0], u2_2Source, u2_2Target)
```

Finally, we have four 3-unitor families in dimension 3, coming from inserting I into the list (A, B, C) at any of the four possible spots. For a strict symmetric monoidal 2-category, each of these is an axiom – in a strict symmetric monoidal 3-category, they would instead each form a new primitive family comprising part of the data.

In this case, there are always three distinct paths mapping the given source to $A \otimes B \otimes C$; we make use of a helper function to select a source and target that can be filled by distinct dimension two instances:

```
c = ConstPrim0("c")
sourceDim0 = comp0s(unit, a, b, c)
targetDim0 = comp0s(a, b, c)
paths = findPaths1(sourceDim0, targetDim0, [u1_L, u1_R])

for ii in range(len(paths)):
    for jj in range(len(paths)):
           if ii != jj:
              print("(%d, %d) - %d paths" % (ii, jj,
                  len(findPaths2(paths[ii], paths[jj], [u2_0, u2_1,
                  u2_2]))))
# >>> (0, 1) - 1 paths
# >>> (0, 2) - 2 paths
# >>> (1, 0) - 0 paths
# >>> (1, 2) - 1 paths
# >>> (2, 0) - 0 paths
# >>> (2, 1) - 0 paths
str(paths[0])
# >>> '[(unitor1L((a @ b @ c)))]'
str(paths[2])
# >>> '[(unitor1L(a) @ (b @ c))]'
```

Thus we see that there is a unique choice of dimension 1 source and target such that there will be exactly two distinct paths between them using the dimension 2 families. We use these as the source and target of our first dimension 3 family.

```
pathsDim2 = findPaths2(paths[0], paths[2], [u2_0, u2_1, u2_2])
str(pathsDim2[0])
# >>> '[[unitor2_0(a, (b @ c))]]'
str(pathsDim2[1])
# >>> '[[unitor2_0((a @ b), c)] & [(unitor2_0(a, b) @ c)]]'


u3_0Source = minimalASTFromEqAEMol2(pathsDim2[0], [a, b, c])
u3_0Target = minimalASTFromEqAEMol2(pathsDim2[1], [a, b, c])
u3_0 = PrimitiveFamily("unitor3_0", 3, 0, [0, 0, 0], u3_0Source,
    u3_0Target)
```

The other dimension 3 families can be defined analogously. To explain the strangeness of their only existing paths for certain choices of of 1-source and 1-target, we note that the definition of symmetric monoidal bicategory requires that each 2-unitor is part of an invertible modification. We provide a family of "inverse" cells for a given family under the `adj` property. Rerunning the path search with these inverse cells included reveals that there are indeed two paths between any choice of 1-source and 1-target:

```
for ii in range(len(paths)):
    for jj in range(len(paths)):
            if ii != jj:
                print("(%d, %d) - %d paths" % (ii, jj,
                    len(findPaths2(paths[ii], paths[jj], [u2_0, u2_1, u2_2,
                    u2_0.adj, u2_1.adj, u2_2.adj]))))
(0, 1) - 2 paths
(0, 2) - 2 paths
(1, 0) - 2 paths
(1, 2) - 2 paths
(2, 0) - 2 paths
(2, 1) - 2 paths
```

We can verify that the directed graph formed by the above 1-cells with edges given by single atoms using 2-unitor primitives is topologically a circle. Thus we can give an equivalent presentation of the 3-unitor axiom by any decomposition of the this circle graph into two halves.

3.5 Extending definitions: χ and ω family

In this section, we'll exhibit an extension to the data and axiom family mentioned at the start of Chapter 2. This extension involves turning the top-level axiom into a piece of data (as a primitive family), and then producing a new top-level axiom involving it by finding fillings of a natural choice of boundary cells.

First, we define a primitive family for $\chi : H(a) \otimes H(b) \to H(a \otimes b)$ as follows.

```
a, b, c, d, e = ConstPrim0("a"), ConstPrim0("b"), ConstPrim0("c"),
    ConstPrim0("d"), ConstPrim0("e")
H = Functor("H")


# We add the functor H in the third argument, indicating that it is a
    functor variable.
chiSource = minimalASTFromMol0(comp0(app(H, a), app(H, b)), [a, b], [H])
chiTarget = minimalASTFromMol0(ensureMol0(app(H, comp0(a, b))), [a, b],
    [H])
# We also set the third argument of chi to 1, indicating that its source
    and target contain a single functor variable.
chi = PrimitiveFamily("chi", 1, 1, [0,0], chiSource, chiTarget)
```

This illustrates another feature of the library: functors. We can think of a functor H as providing a new family of primitives in each dimension, indexed by the cells of that dimension. `CatComputad` contains features for pattern-matching on functors and finding paths underneath of functors.

Next we define a primitive family for ω, which we will now call ω_2. We can find its 1-source and target from fillings, but we also write them down explicitly.

```
Ha, Hb, Hc, Hd, He = app(H,a), app(H,b), app(H,c), app(H,d), app(H,e)
```

```
omega2Source0 = enusreMol0(comp0s(Ha, Hb, Hc))
omega2Target0 = ensureMol0(app(H, comp0s(a, b, c)))
omega2Paths1 = findPaths1(omega2Source0, omega2Target0, [chi])

lmap(str, omega2Paths1)
# >>> ['[(chi{H}(a, b) @ H(c)) . (chi{H}((a @ b), c))]', '[(H(a) @
   chi{H}(b, c)) . (chi{H}(a, (b @ c)))]']

# The .fprimf method is used to get an instance of the primitive family
   with the given parameters.
omega2Source1AST = minimalASTFromEqMol1(ensureEqMol1(
    comp1(
        comp0(Ha, chi1.fprimf([H], b, c)),
        chi1.fprimf([H], a, comp0(b, c)))), [a, b, c], [H])
omega2Target1AST = minimalASTFromEqMol1(ensureEqMol1(
    comp1(
        comp0(chi1.fprimf([H], a, b), Hc),
        chi1.fprimf([H], comp0(a, b), c))), [a, b, c], [H])
omega2 = PrimitiveFamily("omega2", 2, 1, [0, 0, 0], omega2Source1AST,
   omega2Target1AST)
```

We perform the same song and dance to recover the diagrams for the axiom; we will turn this into a new primitive family which we'll call ω_3. Here we face the same difficulty as in Section 3.4, that there are more than two 1-cell paths that can act as source/target. We perform a similar search process, which we omit.

```
omega3Source0 = ensureMol0(comp0s(Ha, Hb, Hc, Hd))
omega3Target0 = ensureMol0(app(H, comp0s(a, b, c, d)))
omega3Paths1 = findPaths1(omega3Source0, omega3Target0, [chi])
```

```
len(omega3Paths1)
# >>> 5


# Some filtering later...
# Source: [((H(a) @ H(b)) @ chi1{0}(c, d)) . (H(a) @ chi1{0}(b, (c @ d))
    . (chi1{0}(a, (b @ c @ d)))]
# --
# Target: [(chi1{H}(a, b) @ (H(c) @ H(d))) . (chi1{H}((a @ b), c) @ H(d))
    . (chi1{0}((a @ b @ c), d))]
omega3Source1 = ensureEqMol1(
    comp1s(
        comp0s(Ha, Hb, chi.fprimf([H], c, d)),
        comp0(Ha, chi.fprimf([H], b, comp0(c, d))),
        chi.fprimf([H], a, comp0s(b, c, d))))
omega3Target1 = ensureEqMol1(
    comp1s(
        comp0s(chi.fprimf([H], a, b), Hc, Hd),
        comp0(chi.fprimf([H], comp0(a, b), c), Hd),
        chi.fprimf([H], comp0s(a, b, c), d)))
omega3Paths2 = findPaths2(omega3Source1, omega3Target1, [omega2])
len(omega3Paths2)
# >>> 2
# omega3 = ...
```

To find the axiom involving omega3, we need to do this again at one higher level. Again, we're faced with the problem of selecting appropriate boundaries, this time in both dimensions 1 and 2. We omit the code, but after some searching we arrive at the diagrams in Fig. 3.1 and Fig. 3.2

3.6 Discussion and future directions

Above, we made mention of the problem of selecting lower-dimensional sources and targets from a number of options when attempting to extend definitions. In the cases we explored, the number of options was still quite limited, such that a full search revealed ones that would lead to successful fillings. For some families, the number of options becomes very large; for instance, there should be a 4-dimensional axiom corresponding to a Breen polytype (as an extension of the family described in [SP09]). The 0-source and 0-target for such an axiom have 38 potential length 2 1-paths between them via the appropriate families, a single pair of which has 792 2-paths between them. Thus the problem can become computationally difficult without other insights to narrow down the 1-paths and 2-paths.

Another difficulty is in knowing exactly which cells are necessary to fill the desired boundaries. In the definition of a symmetric monoidal bicategory, the dimension 1 data comes as bitransformations equipped with adjoint equivalences, while the dimension 2 data comes as invertible modifications. We then expect that for a symmetric monoidal tricategory (or any stricter version thereof) the dimension 1 data should come as tritransformations equipped with an appropriate higher dimensional version of adjoint equivalence, and the dimension 2 data should come as trimodifications equipped with (standard) adjoint equivalences. This leads to an abundance of primitive families: the original family, its adjoint, structure cells on the adjoints (units, counits, and potentially 3-cells replacing the triangle identities), higher dimensional transfor cells, and *their* adjoints and structure cells. Additionally, some of the standard presentations of symmetric monoidal bicategory axioms involve the use of mates. When combined, some combinations of these cells seem to yield explosive growth in the number of available paths.

One might hope that there is a selection of orientations for each cell group such that no adjoints need be involved in the source and target diagrams, eliminating all adjoints and their structure cells from the picture. Indeed, some of the families for symmetric monoidal bicategories usually described using adjoints have speculatively equivalent definitions without. This is a subject for future exploration.

We experience some fundamental difficulties when dealing with 3-cell paths; first and foremost, we have no simple description of 3-cell equivalence via transposition sequences. Thus we can not automatically recognize distinct 3-cell paths as equivalent. Furthermore, when dealing with 3-cells as part of a definition, they will likely be themselves subject to axioms, such as swallowtail axioms as part of a tritransformation adjunction; we have no computational method for applying such axioms, since we don't have the decomposition theorems in dimension 3 needed to use Algorithm 2.3. Thus, 3-cell output currently has to be dealt with "by hand". This suggests an obvious future direction: can such results generalize to strict monoidal 3-computads, or even n-computads?

Another potential direction is to generalize our results to a larger class of 3-computads; recall that a strict monoidal 2-computad is a 3-computad with a single 0-cell. Such results would hopefully be of independent interest for algorithms involving low dimensional computads.

We mentioned above that a certain search space involving unitors seemed to be topologically a circle; the topology of such search spaces could potentially be explored via computer homology computations.

$$\begin{tikzcd}
\overline{a}\overline{b}\overline{c}\chi_{d,e}\cdot\overline{a}\overline{b}\chi_{c,de}\cdot\chi_{a,b}\overline{cde}\cdot\chi_{ab,cde} \arrow[r, equal] & \overline{a}\overline{b}\overline{c}\chi_{d,e}\cdot\chi_{a,b}\overline{c}\overline{de}\cdot\overline{ab}\chi_{c,de}\cdot\chi_{ab,cde} \arrow[r, "1\cdot 1\cdot\omega_{ab,c,de}"] & \overline{a}\overline{b}\overline{c}\chi_{d,e}\cdot\chi_{a,b}\overline{c}\overline{de}\cdot\chi_{ab,c}\overline{de}\cdot\chi_{abc,de} \arrow[r, equal] & \chi_{a,b}\overline{c}\overline{d}\overline{e}\cdot\chi_{ab,c}\overline{d}\overline{e}\cdot\overline{abc}\chi_{d,e}\cdot\chi_{abc,de} \arrow[dr, "1\cdot 1\cdot\omega_{abc,d,e}"] & \\
\overline{a}\overline{b}\overline{c}\chi_{d,e}\cdot\overline{a}\overline{b}\chi_{c,de}\cdot\overline{a}\chi_{b,cde}\cdot\chi_{a,bcde} \arrow[u, "1\cdot 1\cdot\omega_{a,b,cde}"] \arrow[dr, "1\cdot\overline{a}\omega_{b,c,de}\cdot 1"] \arrow[dd, "\overline{a}\overline{b}\omega_{c,d,e}\cdot 1\cdot 1"'] & \Downarrow 1\cdot\Omega_{a,b,c,de} & \overline{a}\overline{b}\overline{c}\chi_{d,e}\cdot\overline{a}\chi_{b,c}\overline{de}\cdot\chi_{a,bc}\overline{de}\cdot\chi_{abc,de} \arrow[u, "1\cdot\omega_{a,b,c}\overline{de}\cdot 1"] \arrow[dr, equal] & = & \chi_{a,b}\overline{c}\overline{d}\overline{e}\cdot\chi_{ab,c}\overline{d}\overline{e}\cdot\chi_{abc,d}\overline{e}\cdot\chi_{abcd,e} \\
& \overline{a}\overline{b}\overline{c}\chi_{d,e}\cdot\overline{a}\chi_{b,c}\overline{de}\cdot\overline{a}\chi_{bc,de}\cdot\chi_{a,bcde} \arrow[ur, "1\cdot 1\cdot\omega_{a,bc,de}"] \arrow[d, equal] & = & \overline{a}\chi_{b,c}\overline{d}\overline{e}\cdot\chi_{a,bc}\overline{d}\overline{e}\cdot\overline{abc}\chi_{d,e}\cdot\chi_{abc,de} \arrow[d, equal] \arrow[dr, "1\cdot 1\cdot\omega_{abc,d,e}"] & \\
\overline{a}\overline{b}\chi_{c,d}\overline{e}\cdot\overline{a}\overline{b}\chi_{cd,e}\cdot\overline{a}\chi_{b,cde}\cdot\chi_{a,bcde} \arrow[d, "1\cdot\overline{a}\omega_{b,cd,e}\cdot 1"'] \arrow[dr, phantom, "\Downarrow\ \overline{a}\Omega_{b,c,d,e}\cdot 1"] & \overline{a}\chi_{b,c}\overline{d}\overline{e}\cdot\overline{a}\overline{bc}\chi_{d,e}\cdot\overline{a}\chi_{bc,de}\cdot\chi_{a,bcde} \arrow[rr, "1\cdot 1\cdot\omega_{a,bc,de}"] \arrow[dr, "1\cdot\overline{a}\omega_{bc,d,e}\cdot 1"] & & \overline{a}\chi_{b,c}\overline{d}\overline{e}\cdot\overline{a}\overline{bc}\chi_{d,e}\cdot\chi_{a,bc}\overline{de}\cdot\chi_{abc,de} \arrow[d, phantom, "\Downarrow\ 1\cdot\Omega_{a,bc,d,e}"] & \overline{a}\chi_{b,c}\overline{d}\overline{e}\cdot\chi_{a,bc}\overline{d}\overline{e}\cdot\chi_{abc,d}\overline{e}\cdot\chi_{abcd,e} \arrow[uu, "\omega_{a,b,c}\overline{d}\overline{e}\cdot 1\cdot 1"] \\
\overline{a}\overline{b}\chi_{c,d}\overline{e}\cdot\overline{a}\chi_{b,cd}\overline{e}\cdot\overline{a}\chi_{bcd,e}\cdot\chi_{a,bcde} \arrow[rr, "\overline{a}\omega_{b,c,d}\overline{e}\cdot 1\cdot 1"] & & \overline{a}\chi_{b,c}\overline{d}\overline{e}\cdot\overline{a}\chi_{bc,d}\overline{e}\cdot\overline{a}\chi_{bcd,e}\cdot\chi_{a,bcde} \arrow[r, "1\cdot 1\cdot\omega_{a,bcd,e}"] & \overline{a}\chi_{b,c}\overline{de}\cdot\overline{a}\chi_{bc,d}\overline{e}\cdot\chi_{a,bcd}\overline{e}\cdot\chi_{abcd,e} \arrow[ur, "1\cdot\omega_{a,bc,d}\overline{e}\cdot 1"] &
\end{tikzcd}$$

Figure 3.1: The first 3-cell pasting diagram for the ω family level 4 axiom. $\overline{a}$ is shorthand for $H(a)$, adjacency indicates tensor, and $\cdot$ indicates horizontal composition. Ω is used for the dimension 3 family. Vertices are 1-cells and edges are 2-cells.

$$\begin{tikzcd}
\bar a\bar b\bar c\chi_{d,e}\cdot\bar a\bar b\chi_{c,de}\cdot\chi_{a,b}\bar c\overline{de}\cdot\chi_{ab,cde} \arrow[r, equal] \arrow[ddr, "\bar a\bar b\omega_{c,d,e}\cdot1\cdot1"] & \bar a\bar b\bar c\chi_{d,e}\cdot\chi_{a,b}\bar c\overline{de}\cdot\overline{ab}\chi_{c,de}\cdot\chi_{ab,cde} \arrow[r, "1\cdot1\cdot\omega_{ab,c,de}"] & \bar a\bar b\bar c\chi_{d,e}\cdot\chi_{a,b}\bar c\overline{de}\cdot\chi_{ab,c}\overline{de}\cdot\chi_{abc,de} \arrow[r, equal] & \chi_{a,b}\bar c\bar d\bar e\cdot\chi_{ab,c}\bar d\bar e\cdot\overline{abc}\chi_{d,e}\cdot\chi_{abc,de} \arrow[dr, "1\cdot1\cdot\omega_{abc,d,e}"] & \\
\bar a\bar b\bar c\chi_{d,e}\cdot\bar a\bar b\chi_{c,de}\cdot\bar a\chi_{b,cde}\cdot\chi_{a,bcde} \arrow[u, "1\cdot1\cdot\omega_{a,b,cde}"] \arrow[dd, "\bar a\bar b\omega_{c,d,e}\cdot1\cdot1"'] & = & 1\cdot\Omega_{ab,c,d,e}\ \Downarrow & & \chi_{a,b}\bar c\bar d\bar e\cdot\chi_{ab,c}\bar d\bar e\cdot\chi_{abc,d}\bar e\cdot\chi_{abcd,e} \\
& \bar a\bar b\chi_{c,d}\bar e\cdot\bar a\bar b\chi_{cd,e}\cdot\chi_{a,b}\overline{cde}\cdot\chi_{ab,cde} \arrow[r, equal] \arrow[d, equal] & \chi_{a,b}\bar c\bar d\bar e\cdot\overline{ab}\chi_{c,d}\bar e\cdot\overline{ab}\chi_{cd,e}\cdot\chi_{ab,cde} \arrow[r, "1\cdot1\cdot\omega_{ab,cd,e}"] \arrow[d, phantom, "="] & \chi_{a,b}\bar c\bar d\bar e\cdot\overline{ab}\chi_{c,d}\bar e\cdot\chi_{ab,cd}\bar e\cdot\chi_{abcd,e} \arrow[ur, "1\cdot\omega_{ab,c,d}\bar e\cdot1"] \arrow[dl, equal] & \\
\bar a\bar b\chi_{c,d}\bar e\cdot\bar a\bar b\chi_{cd,e}\cdot\bar a\chi_{b,cde}\cdot\chi_{a,bcde} \arrow[ur, "1\cdot1\cdot\omega_{a,b,cde}"] \arrow[d, "1\cdot\bar a\omega_{b,cd,e}\cdot1"'] & \bar a\bar b\chi_{c,d}\bar e\cdot\chi_{a,b}\overline{cd}\bar e\cdot\overline{ab}\chi_{cd,e}\cdot\chi_{ab,cde} \arrow[r, "1\cdot1\cdot\omega_{ab,cd,e}"] \arrow[d, phantom, "\Downarrow\ 1\cdot\Omega_{a,b,cd,e}"] & \bar a\bar b\chi_{c,d}\bar e\cdot\chi_{a,b}\overline{cd}\bar e\cdot\chi_{ab,cd}\bar e\cdot\chi_{abcd,e} & \Omega_{a,b,c,d}\bar e\cdot1\ \Downarrow & \bar a\chi_{b,c}\bar d\bar e\cdot\chi_{a,bc}\bar d\bar e\cdot\chi_{abc,d}\bar e\cdot\chi_{abcd,e} \arrow[uu, "\omega_{a,b,c}\bar d\bar e\cdot1\cdot1"'] \\
\bar a\bar b\chi_{c,d}\bar e\cdot\bar a\chi_{b,cd}\bar e\cdot\bar a\chi_{bcd,e}\cdot\chi_{a,bcde} \arrow[rr, "1\cdot1\cdot\omega_{a,bcd,e}"'] & & \bar a\bar b\chi_{c,d}\bar e\cdot\bar a\chi_{b,cd}\bar e\cdot\chi_{a,bcd}\bar e\cdot\chi_{abcd,e} \arrow[u, "1\cdot\omega_{a,b,cd}\bar e\cdot1"'] \arrow[r, "\bar a\omega_{b,c,d}\bar e\cdot1\cdot1"] & \bar a\chi_{b,c}\bar d\bar e\cdot\bar a\chi_{bc,d}\bar e\cdot\chi_{a,bcd}\bar e\cdot\chi_{abcd,e} \arrow[ur, "1\cdot\omega_{a,bc,d}\bar e\cdot1"] &
\end{tikzcd}$$

Figure 3.2: The second 3-cell pasting diagram for the ω family level 4 axiom. Note the bottom two 2-cells appear in a different order than the first diagram, but commute past each other.

Bibliography

[BMMS00] J.-C. Birget, S. Margolis, J. Meakin, and M. Sapir, editors. *Algorithmic problems in groups and semigroups*. Trends in Mathematics. Birkhäuser Boston, Inc., Boston, MA, 2000. Papers from the International Conference held at the University of Nebraska, Lincoln, NE, May 11–16, 1998.

[GJO17] N. Gurski, N. Johnson, and A. M. Osorno. The 2-dimensional stable homotopy hypothesis. *ArXiv e-prints*, December 2017.

[JO12] Niles Johnson and Angélica M. Osorno. Modeling stable one-types. *Theory Appl. Categ.*, 26:No. 20, 520–537, 2012.

[JS93] A. Joyal and R. Street. Braided tensor categories. *Advances in Mathematics*, 102(1):20 – 78, 1993.

[Mak05] Michael Makkai. The word problem for computads, 2005.

[SP09] Christopher John Schommer-Pries. *The classification of two-dimensional extended topological field theories*. ProQuest LLC, Ann Arbor, MI, 2009. Thesis (Ph.D.)–University of California, Berkeley.