university of groningen

faculty of science and engineering

Almost Invariants for Framed and Spin 3-Manifolds via Hopf Algebras

Master Project Mathematics

August 2025

Student: B.B. Tielman

First supervisor: Prof.dr. R.I. van der Veen

Second supervisor: Dr. T.F. Görbe

Abstract

In 2021, Serban Matei Mihalache, Sakie Suzuki and Yuji Terashima submitted a paper that constructs invariants of closed 3-manifolds using involutory Hopf algebras. In this paper we will follow their construction without assuming the Hopf algebra is involutory. Instead, we will assume there exist some invertible involutive element $u \in H^*$ such that applying the dual antipode squared is equal to conjugation by u. i.e. $(S^*)^2(x) = u^{-1}xu$, and $u = u^{-1}$. We get an almost invariant of framed and spin manifolds. Our construction will be invariant up to the $\Omega 0$ move.

Contents

1 Introduction **2**

2 Framed and spin 3-manifolds **4**

3 O-graph axioms **4**
3.1 Equivalent o-graphs . . . 6

4 O-graphs construction **8**
4.1 Spine . . . 8
4.2 Branching . . . 9
4.3 Example of constructing an o-graph . . . 10

5 Hopf Algebras **11**
5.1 Dual Hopf algebra . . . 14
5.2 Pivots, integrals and cointegrals . . . 14
5.3 The Heisenberg Double . . . 17
5.4 Assumptions on H and H^* . . . 17
5.5 The Radford Algebra . . . 17

6 Construction of the invariant **19**
6.1 Fock representation . . . 20
6.2 Tensor diagram . . . 20
6.3 Continuation of construction . . . 23
6.4 Examples of the invariant . . . 24

7 Proof of invariance **26**
7.1 The 0-2 move . . . 26
7.2 The MP moves . . . 27
7.3 The Pontrjagin move . . . 30

8 Other representations **37**

9 Differences with previous findings and future recommendations **38**

10 Appendix: Mathematica code **40**
10.1 Initialization cell text from . . . 40
10.2 Other lines to run text form . . . 48
10.3 Initialization cell image form . . . 52
10.4 Other lines to run image form . . . 58

1 Introduction

Around the 1960s, the first glimpses of what later became o-graphs already appeared in the form of special spines of 3-manifolds. In 1965, Casler already wrote about standard spines of manifolds [3]. Over the many years, people have done more study about them, with some names like Matveev, Piergallini and Ishii making broad advancements within the study of spines and thus o-graphs. The MP moves, for example, are also named after Matveev and Piergallini. At this stage, o-graphs were undirected, but this changed with the introduction of a branching. The idea of a branching on spines resulted in the book "Branched standard spines of 3-manifolds" by Benedetti and Petronio.

Way before all of this, in 1941, Heinz Hopf wrote a paper about algebraic topology. This is where the notion of a Hopf algebra was born, mainly to deal with representations of Lie algebras. Here too a lot of people worked on this field. With Sweedler's book "Hopf Algebras," the field began to take proper shape [1], with people like Radford and Kuperberg following after. Nowadays, Hopf algebras are widely used in knot theory and representation theory.

In more recent history, people have been combining these two branches of mathematics to find invariants of 3-manifolds. This resulted in the 2021 paper by Matei Mihalache, Sakie Suzuki and Yuji Terashima [7]. This paper, together with the above mentioned works has formed the basis of this research, in which we wished to generalize the results from Mihalache, Suzuki and Terashima. They found an invariant of 3-manifolds by using involutory Hopf algebras. Guided by our intuition and knowledge of knot theory, we thought we could do away with the assumption of the Hopf algebra being involutory. In doing so, we had to impose other restrictions, which has led us to a presumably new (almost) invariant for spin 3-manifolds.

2 Framed and spin 3-manifolds

We will work with oriented, compact, connected, three dimensional manifolds M with empty boundary. We will sometimes also say closed, which means compact and with empty boundary. We will consider these manifolds up to orientation preserving diffeomorphism.

A framing on a 3-manifold is a choice of three linearly independent vector fields that pointwise induce the orientation. A manifold with a chosen framing is called a framed manifold. A framing is considered up to homotopy.

A spin structure on a manifold M is a framing on M with the following property: if P is a spine of M and $S(P)$ its set of triple lines, then the framing on $S(P)$ should extend to a framing on P. These framings are considered up to homotopy on $S(P)$. This is the definition given by Benedetti and Petronio given in [2] on page 94. The definitions of P and $S(P)$ can be found later at the end of Section 4.1.
For a different reference on spin, see "Spin Structures on Manifolds" by Milnor [8].

3 O-graph axioms

Different types of manifolds can all be encoded by various o-graphs. If the type of o-graph is clear from the context, we will often drop the adjective that describes the type, and simply call it an o-graph, instead of a framed/spin o-graph. All these definitions can be found in Benedetti and Petronio's book [2]. We are mainly interested in spin o-graphs, but our invariant works for framed o-graphs as well, because framed o-graphs can be regarded as a representative element of an equivalence class. This class can then be seen as a spin o-graph.

We will now give the definitions for framed and spin o-graphs.

Definition 1 (Framed o-graph)**.** A framed o-graph is a graph Γ where every vertex has degree 4, together with extra structure. The structure will be described in the axioms. They are:

- N1.) The graph Γ has at least one vertex. For every vertex, we pair up the two opposite edges and mark one as going over the other. We depict these vertices in the plane with a dot.
- N2.) Each edge has a direction and the directions of opposite edges through a vertex match.
- C1.) If we remove the vertices of the graph and join the opposite edges together, we get a single unique oriented circuit.
- C2.) If we change the graph by the moves in Figure 1, the resulting trivalent graph should be connected.
- C3.) If we apply the moves in Figure 2, we should get a disjoint union of circuits. The number of these circuits is exactly one more than the number of vertices of the original o-graph Γ.
- F1.) There is an element of $\mathbb{Z}/2\mathbb{Z}$ attached to each edge, which is called the color.
- F2.) It is possible to attach an integer x_i to each edge e_i that is congruent to the color modulo 2. These integers satisfy the following. For each circuit γ that we obtain by the moves of Figure 2 we retrace the path in our original o-graph Γ. We should have that there are $2(1 - \sum \alpha_i x_i)$ solid dots on γ, where $\alpha_i \in \{-1, 1\}$ denotes the orientation of the transfersal of the path in Γ, i.e. $e_i^{\alpha_i}$ in Γ corresponds to a piece of the circuit γ.

A spin o-graph is defined relatively similarly. Just as a spin structure is a bit more general than a framing, so is a spin o-graph a bit more general than a framed o-graph.

Definition 2 (Spin o-graph)**.** A spin o-graph is a graph Γ where every vertex has degree 4, together with the following axioms:

- N1.) The graph Γ has at least one vertex. For every vertex, we pair up the two opposite edges and mark one as going over the other. We depict these vertices in the plane with a dot.
- N2.) Each edge has a direction and the directions of opposite edges through a vertex match.
- C1.) If we remove the vertices of the graph and join the opposite edges together, we get a single unique oriented circuit.
- C2.) If we change the graph by the moves in Figure 1, the resulting trivalent graph should be connected.
- C3.) If we apply the moves in Figure 2, we should get a disjoint union of circuits. The number of these circuits is exactly one more than the number of vertices of the original o-graph Γ.

- S1.) There is an element of $\mathbb{Z}/2\mathbb{Z}$ attached to each edge, which is called the color.
- S2.) Let γ be a circuit obtained by doing the replacement of Figure 2. If there are $2n$ solid dots on the circuit, then $\sum_i x_i \equiv n+1 \mod 2$, where x_i is the color on the edge e_i.

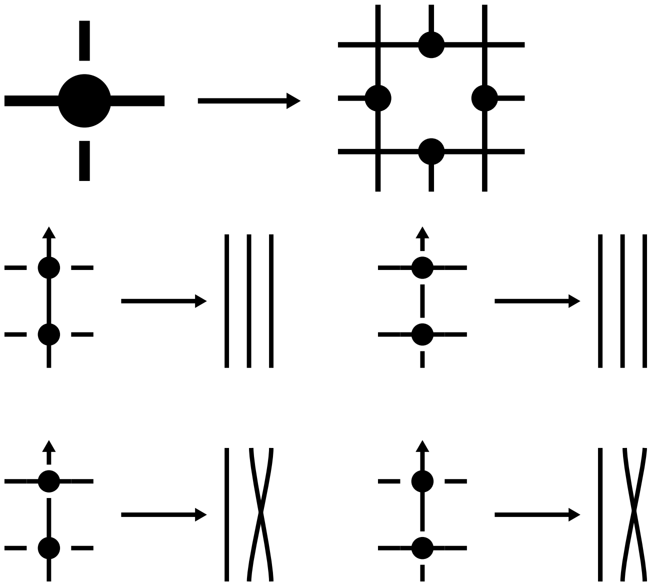


Figure 1: The replacement that one needs to do to check for C2.

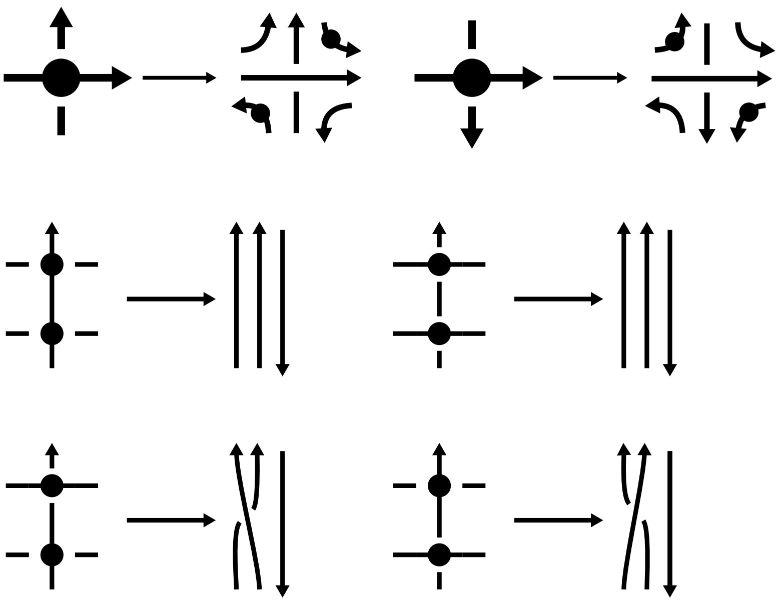


Figure 2: The other replacement to check C3.

The diagrams in Figures 1 and 2 should be read in the following way: whenever we have a piece of strand, we look at the direction that the strand has. At the beginning and end of the strand there is a crossing where the strand goes either over or under. The replacement that we do depends on over-/under-ness of the strands with the crossing. If for example the strand goes over the crossing in both the beginning and end, then you just replace the strand by 3 lines that do not cross.

We want to depict these o-graphs in the plane, but often the o-graphs are not planar. We will therefore depict these o-graphs by showing real crossings with a dot and so called phantom crossings without a dot. Any diagram should be viewed up to planar isotopy. Because of the embedding in the plane and phantom crossings, we also get the Reidemeister-like moves as seen in Figure 3.

In Figure 3 we have to understand that there are directions and colors on the strands. When you apply the moves of the diagram, the directions and colors have to stay consistent.

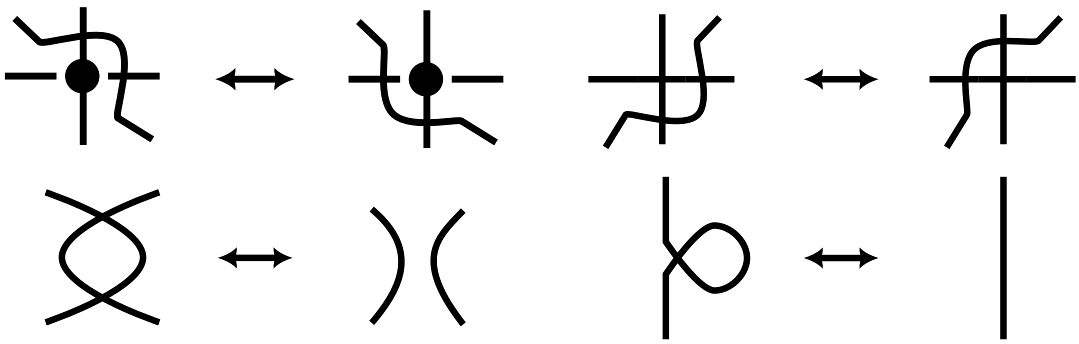


Figure 3: Reidemeister-like moves for phantom crossings.

Example 3. An example of a spin o-graph is that of a 3-sphere $\mathbb{S}^3 = \{x \in \mathbb{R}^4 : ||x|| = 1\}$ with a certain framing. The normal o-graph can be seen in Figure 4. We can check that this does satisfy all axioms. This example and the next example are taken from [6].

Example 4. A different example for a spin o-graph is an o-graph of the Lens space $L(2,1)$. This and the pervious example were obtained from “Quantum Invariants of Closed Framed 3-Manifolds Based on Ideal Triangulations” [6], by taking the framing modulo 2. We then get a spin structure for it. This spin o-graph of $L(2,1)$ is shown in Figure 5. This particular Lens space is equal to the real projective space $\mathbb{RP}^3$.

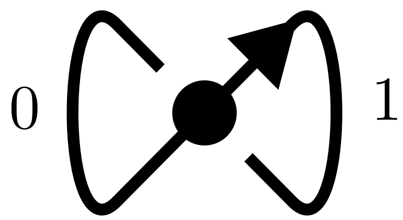


Figure 4: A spin o-graph of $\mathbb{S}^3$.

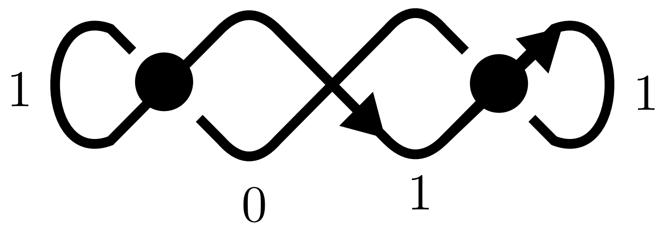


Figure 5: A spin o-graph of a spin $L(2,1)$, which is the real projective space $\mathbb{PR}^3$.

3.1 Equivalent o-graphs

In Benedetti and Petronio’s book [2] it is said in a roundabout way in Theorem 1.4.3 and Theorem 1.4.4 that the set of framed o-graphs under some equivalence relations correspond bijectively to framed manifolds. Similarly, the set of spin o-graphs under (a slightly different) equivalence relations correspond bijectively to spin manifolds. This means that if one comes up with a construction that takes an o-graph, and that is invariant under the equivalence relations, then one has found an invariant of spin / framed manifolds.

The equivalence relation on o-graphs for framed manifolds are the following three: the 0-2 move, the MP moves, and the Ω0 move. For spin o-graphs there are four relations: the 0-2 move, the MP moves, the 3-5 move and Ω0. The 3-5 move is often also called the Pontrjagin move. One can find pictures for all these moves in Figures 6, 7, 8, and 9.

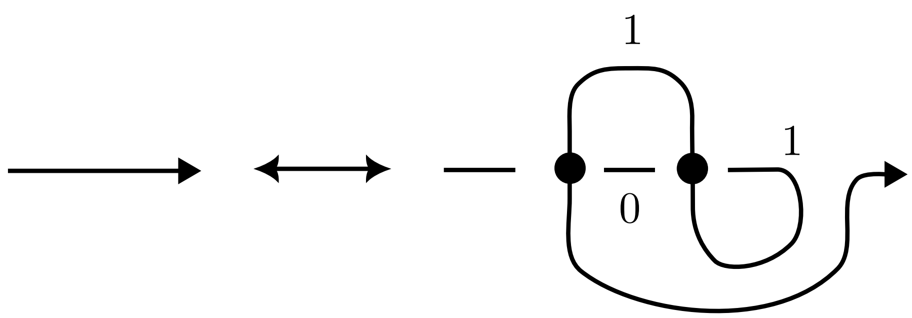


Figure 6: The 0-2 move.

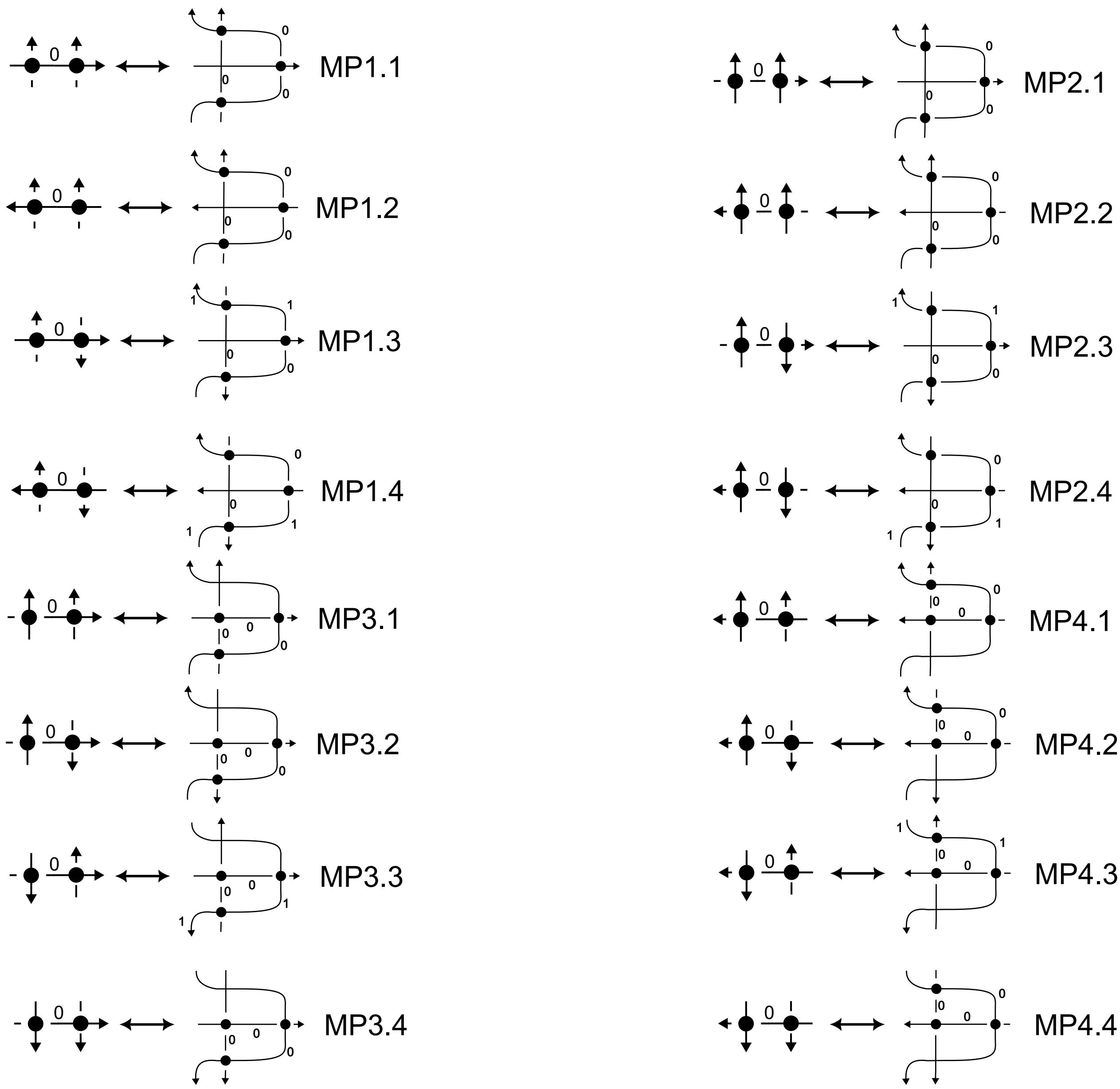


Figure 7: The MP moves.

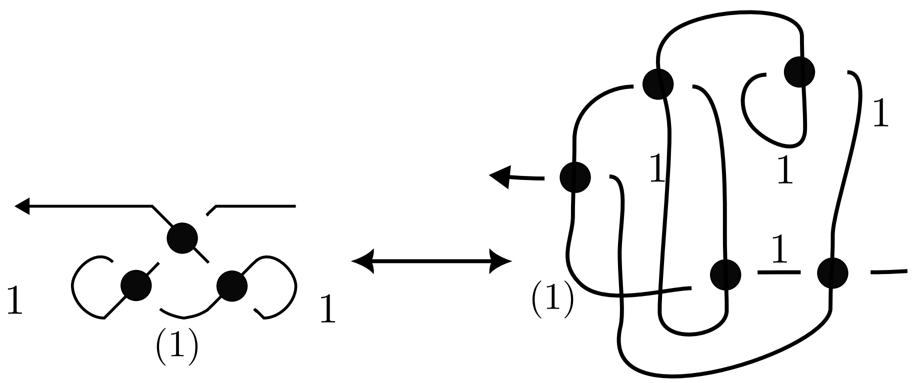


Figure 8: The 3-5 move(s).

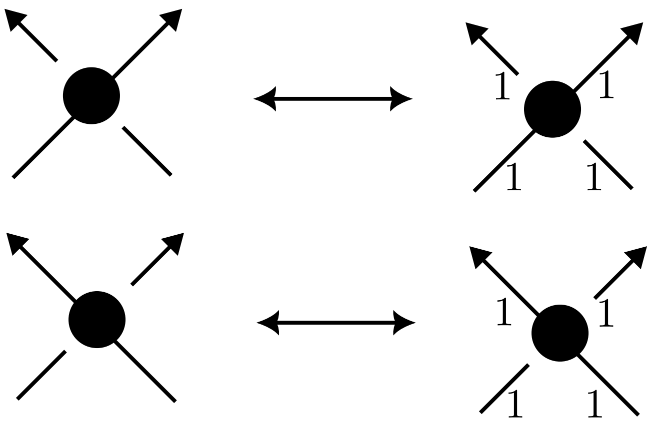


Figure 9: The Ω0 moves.

Some explanation about these moves is needed. Firstly, if an edge is at the beginning or end of a diagram, then it does not show any color. This is because it is convenient to think of color as being localized at a single point of the edge, and able to move around on the edge. We are also allowed to think about color as being localized at multiple points, and then summing the numbers (modulo 2 for spin) to get the final color. This means that edges at the beginning or end of a diagram can carry any color. An exception to this is the Ω0. This move does have color on edges at the beginning and end. This is because the Ω0 move tells us we are allowed to change the local color at a vertex as long as we do it on all four edges at the same time. Secondly, there are two versions of the 3-5 move. This is indicated by the 1 between parentheses. There is a version where that 1 is really there on both the left hand side and the right hand side, and a version where it is not there on both the left hand side and right hand side. It is also important to note that we have omitted the color 0 from edges in the 3-5 move to make the diagram easier to read.
The final remark that one should notice is that the Ω0 move is self-inverse for spin o-graphs in the sense that applying the move twice (both from the left hand side version to the right hand side version) is the same as doing nothing for spin 0-graphs.

4 O-graphs construction

4.1 Spine

How do we obtain an o-graph? There are explicit ways to obtain o-graphs given a 3-manifold. We will give a rough overview of how to obtain an o-graph from a manifold. We will not show how you get from the framing / spin structure to the colors on the edge.
We will start with construction due to Matveev in [5]. We start with abstract symplicial complexes.

Definition 5. An abstract symplicial complex, or ASC, is a finite set K such every element $k \in K$ is again a set. Moreover, for any subset of an element $k' \subseteq k \in K$, we must have that $k' \in K$.

With an abstract symplicial complex we can construct polyhedron, and with polyhedron we can construct spines. If we have an oriented 3-manifold with boundary, then we can always approximate it by an ASC. Benedetti and Petrionio leave this as a remark in chapter 2.1. These ASCs will give us the notion of a spine quite easily. We can collapse ASCs to make them smaller. We therefore introduce the following definitions.

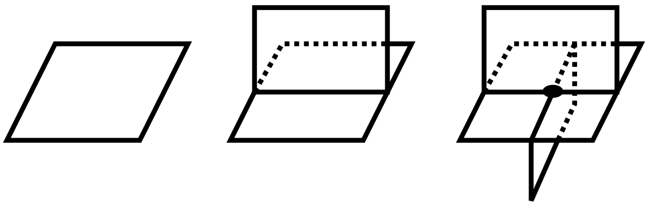


Figure 10: The three possible local images of a spine.

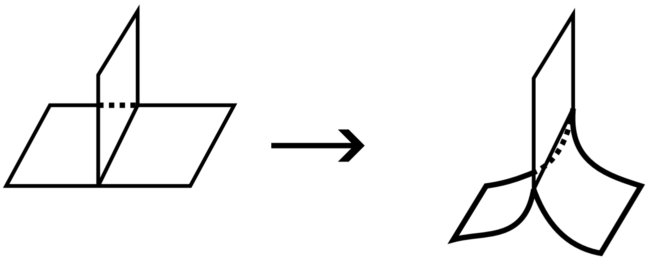


Figure 11: A branched edge.

Definition 6. A free face of an ASC is pair of elements $(f, F) \in K$ with $f \neq \varnothing$ such that f is a proper face of F while not being a proper face of any other element of K.

Definition 7. Given a ASC K and a free face (f, F) of K, we say $L = K \setminus \{f, F\}$ is a simplicial collapse of K. We say that K and L are simply homotopic.

Definition 8. Given ASC K and L, we say that K is simply homotopic to L if there is a finite chain of simplicial collapses that link K and L. We will write $K \approx L$.

We can view these ASCs as a piecewise linear manifold. Being simply homotopic is an equivalence relation. This is not too hard to see. If we have an approximation of an oriented, compact, 3-manifold with boundary by an ASC, then we can do simplicial collapse to make every point locally look like one of these three forms if, we were to embed the ASC in $\mathbb{R}^3$. These three forms are either a plane: $\mathbb{R}^2 \times \{0\}$, a plane with a half plane on top $\big(\mathbb{R}^2 \times \{0\}\big) \cup (\mathbb{R} \times \{0\} \times \mathbb{R}_{\geq 0})$, and the plane with a half plane on the top and bottom $\big(\mathbb{R}^2 \times \{0\}\big) \cup (\mathbb{R} \times \{0\} \times \mathbb{R}_{\geq 0}) \cup (\mathbb{R} \times \mathbb{R}_{\leq 0} \times \{0\}\big)$. Pictures for these can be seen in Figure 10. By doing simplicial collapse until every point of the ASC locally looks like one of these three forms, we have obtained a spine P.
A spine has three different types of points: points that are on a plane, points that are on the intersection of a plane with a half plane, and points that are on the intersection of a plane with two half planes. The second set of points, called triple lines (also called edges), will be denoted by $S(P)$. The third set of points are called true vertices (also called vertices) and are denoted by $V(P)$. They correspond to points that locally look like the middle of the third picture in Figure 10. We also demand that a spine needs to satisfy that $P \setminus S(P)$ are open disks, and $S(P) \setminus V(P)$ open line segments. We can construct these spines for oriented, compact 3-manifolds with non-empty boundary, but we are looking at oriented closed 3-manifolds. For an oriented closed 3-manifold we puncture the manifold first. This means that we remove an open 3-ball somewhere. The o-graph of the oriented closed 3-manifold will thus be given by the spine of the punctured manifold $M \setminus \mathrm{Int}(\mathbb{B}^3)$, where $\mathbb{B}^3$ is a 3-ball in M. If we have found a spine, then we can construct the o-graph. The true vertices of the spine become vertices of the o-graph, and the triple lines tell us how to connect the vertices. But this does not guarantee a branched spine. There are more steps one need to take if one wishes to make a branched spine. For example, it may happen that the found spine gives rise to an o-graph that consists of multiple circuits. We therefore refer for the full construction to Benedetti and Petronio's book [2].

4.2 Branching

The previous section described how to get just any o-graph, but it was not able to explain how an edge gets its direction. This is also done via branching. The easiest way to get an o-graph with direction on the edges is to make the spine out of the two pieces you see in the left hand side of Figure 12. They correspond to positively and negatively oriented vertices. Roughly speaking, a branching makes the planes around a triple line transition smoothly. We then also want such a smooth transition at the vertices. We then try to puzzle the spine together using only these two pieces and by demanding that we do not break this smoothness.
One sees that one now also needs to take into consideration the orientation of the upper and lower half planes to choose the right vertex type.

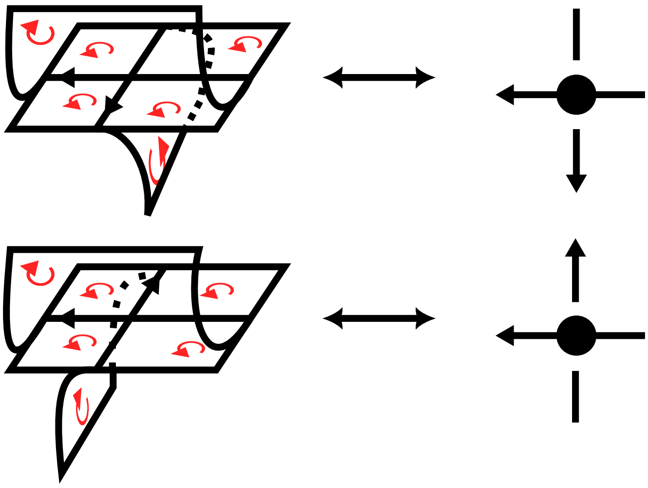


Figure 12: The two possible types of vertices.

4.3 Example of constructing an o-graph

We will give an overview of the construction of the easiest example of an o-graph; a o-graph of a 3-sphere $\mathbb{S}^3$.

First we puncture $\mathbb{S}^3$. Since $\mathbb{S}^3$ is obtained by glueing two 3-balls together along their boundary, we get a 3-ball if we puncture $\mathbb{S}^3$. In other words $\mathbb{S}^3 \setminus \text{Int}(\mathbb{B}^3) = \mathbb{B}^3$. We therefore start off with a solid ball. We will depict the ball as a solid cube. We make an indent on the top and extend this indent up till halfway inside the cube. From here on out we will vacuum up all the space inside the lower half, but we will make sure to leave a pillar. We cannot just leave a pillar, as our vacuuming has to be done in such a way that we will end up with something with no holes, so this means that the remaining pillar is connected to the outside via a wall. We now do the exact same on the bottom of the cube as we did on the top. What we end up with is called the house with two rooms, or Bing's house.

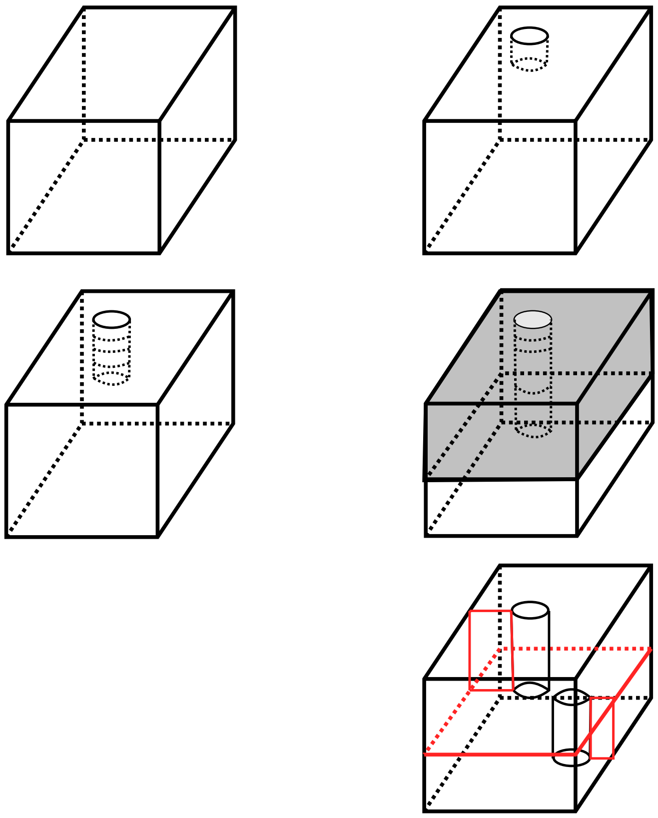


Figure 13: From the solid ball to the house with two rooms. The triple line of the spine has been marked red.

We now want to construct this spine using only the pieces found in Figure 12. Alternatively, one can transform the solid ball into an abalone and obtain the o-graph in Figure 4.

5 Hopf Algebras

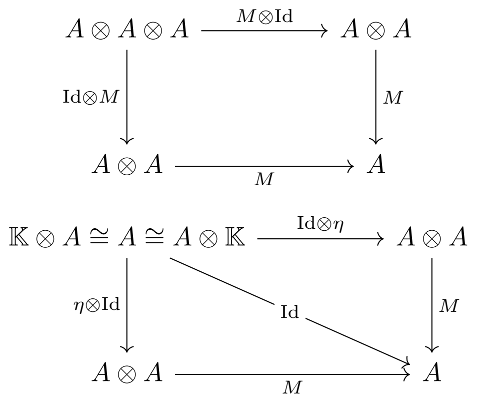
Our main tool that we will use to construct or invariant are Hopf algebras. Like said before, they are rather useful in knot theory. Any normal algebra would be enough to multiply elements together on a decorated diagram, but Hopf algebras are especially nice to work with because of their antipode and coalgebra structure. Loosely speaking, in knot theory, multiplying two elements on a decorated diagram corresponds with joining the strands together in the knot, namely the strands that the given elements represent. Using the comultiplication corresponds to doubling a strand, and using the antipode corresponds to reversing the direction of traversal of a strand.

Of course, o-graphs are not knots, but they resemble and behave just close enough to knots that we can still use Hopf algebras and similar techniques to find invariants.

The invariant that we will construct will be dependent on the chosen Hopf algebra, which also need to have certain additional properties. We will slowly build up toward the definition of a Hopf algebra in this section by first introducing algebras and coalgebras. For reference on Hopf algebras one can see Sweedler's book [12], or Radford's book [10].

Definition 9. A unitary associative algebra A over a field $\mathbb{K}$ (henceforth just called an algebra) is a $\mathbb{K}$-vector space equipped with bilinear map $M : A \otimes A \to A$, called multiplication, and a linear map $\eta : \mathbb{K} \to A$, called the unit, such that the following equations are satisfied.

$$M(M(x,y),z) = M(x, M(y,z)),$$
$$M(\eta(1), x) = x = M(x, \eta(1)).$$



Instead of writing M all the time, we denote M with a dot, and write $M(x,y)$ as $x \cdot y$, or we use concatenation and write xy.

It should be noted that the generic term 'algebra' is sometimes also used to describe an algebra over any commutative ring. We will not do this. Our algebras will always be over fields.

Remark. We will often refer to finite dimensional (Hopf) algebras. The dimension of a (Hopf) algebra is the dimension of the vector space.

The main Hopf algebra that we will be using is the Radford algebra. We'll slowly introduce the full structure of this algebra over this section. The Radford algebra $\mathcal{R}$ is an eight dimensional Hopf algebra. One can construct it as follows. First take the free algebra over the non-commuting indeterminates g, x and y, so $\mathbb{K}\langle g, x, y\rangle$. Now take the ideal generated by $\{g^2 - 1, x^2 - x, y^2 - y, gx + xg - (g-1), gy + yg - (g-1), xy + yx - (x+y)\}$ and take quotient of the free algebra with this ideal. The result is the algebraic part of Radford Hopf algebra, whose multiplication table is depicted in Table 1.

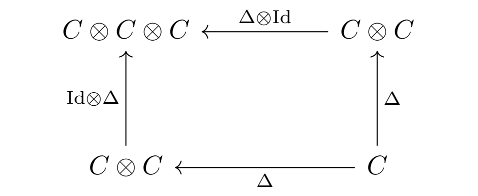
	1	g	x	y	gx	gy	xy	gxy
1	1	g	x	y	gx	gy	xy	gxy
g	g	1	gx	gy	x	y	gxy	xy
x	x	$g-1-gx$	x	xy	$-x$	$gy-y-gxy$	xy	$-xy$
y	y	$g-1-gx$	$x+y-xy$	y	$gxy-x-xy$	$-y$	y	$gxy-gy-xy$
gx	gx	$1-g-x$	gx	gxy	$-gx$	$y-gy-xy$	gxy	$-gxy$
gy	gy	$1-g-y$	$gx+gy-gxy$	gy	$xy-y-gx$	$-gy$	gy	$xy-y-gxy$
xy	xy	$g+y+gxy-1-x-gx-gy$	x	xy	$y+gxy-x-gy-xy$	$-xy$	xy	$gxy+y-gy-2xy$
gxy	gxy	$1+gy+xy-g-x-y-gx$	gx	gxy	$gy+xy-y-gx-gxy$	$-gxy$	gxy	$gy+xy-y-2gxy$

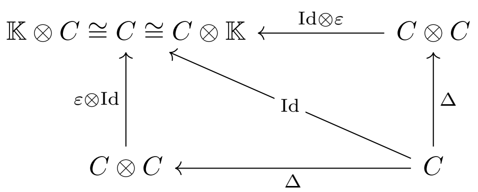
Table 1: Multiplication table for Radford algebra.

We continue with our goal of defining a Hopf algebra. A Hopf algebra is not only a algebra, but also a coalgebra. A coalgebra is defined as follows.

Definition 10. A counitary coassociative coalgebra C over a field $\mathbb{K}$ (henceforth just called a coalgebra) is a $\mathbb{K}$-vector space equipped with a linear map $\Delta : C \to C \otimes C$, called comultiplication, and linear map $\varepsilon : C \to \mathbb{K}$, called the counit, such that the following equations are satisfied.

$$(\mathrm{Id} \otimes \Delta)(\Delta(x)) = (\Delta \otimes \mathrm{Id})(\Delta(x)),$$
$$(\mathrm{Id} \otimes \varepsilon)(\Delta(x)) = x = (\varepsilon \otimes \mathrm{Id})(\Delta(x)).$$





We will often use reduced Sweedler notation. This means that we write $\Delta(x) = \sum x_{(1)} \otimes x_{(2)}$, or even more shortly just $\Delta(x) = x_{(1)} \otimes x_{(2)}$, instead of the full expression $\Delta(x) = \sum_{i=1}^{n} a_i \otimes b_i$ where we then also need to specify what a_i, b_i and n are. Using this notation we can restate coalgebra the axioms as

$$x_{(1)} \otimes (x_{(2)})_{(1)} \otimes (x_{(2)})_{(2)} = (x_{(1)})_{(1)} \otimes (x_{(1)})_{(2)} \otimes x_{(2)},$$
$$x_{(1)} \otimes \varepsilon(x_{(2)}) = x = \varepsilon(x_{(1)}) \otimes x_{(2)}.$$

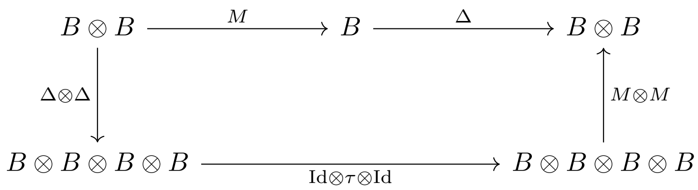
Again we will clarify this via the Radford Hopf algebra. The coalgebra structure on the Radford algebra is defined by $\Delta(1) = 1 \otimes 1, \Delta(g) = g \otimes g$, $\Delta(x) = x \otimes g + 1 \otimes x$ and $\Delta(y) = y \otimes g + 1 \otimes y$. The counit is $\varepsilon(1) = 1, \varepsilon(g) = 1$, $\varepsilon(x) = \varepsilon(y) = 0$. In what follows, we will see that we did not really need to specify the behavior of Δ and ε on 1, since it will follow from the bialgebra axioms. We also have not specified what the outcome of calculations like $\Delta(gx)$ should be, but those also will become clear when we discuss the bialgebra axioms.

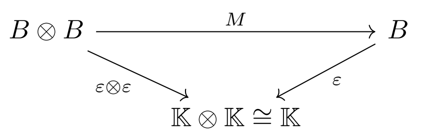
A bialgebra is an object that behaves as an algebra and coalgebra at the same time, and where the operations work together in the way one would expect. The definition is as follows.

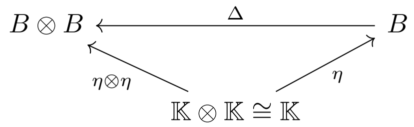
Definition 11. A bialgebra B over a field $\mathbb{K}$ is a $\mathbb{K}$-vector spaces together with the maps $M, \eta, \Delta, \varepsilon$ such that B is an algebra with M and η, and a coalgebra with Δ, ε. Moreover, the following equations should hold for $a, b \in B$, and $r \in \mathbb{K}$.

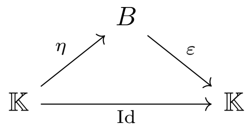
$$\Delta(ab) = a_{(1)}b_{(1)} \otimes a_{(2)}b_{(2)},$$
$$\varepsilon(ab) = \varepsilon(a) \cdot \varepsilon(b),$$
$$\Delta(r) = r(1 \otimes 1),$$
$$\varepsilon(\eta(r)) = r.$$

All these can be expressed by the following commutative diagrams.









In a bialgebra, we have that η is injective. Because of this, we often view $\mathbb{K}$ as a subset of B. This allows us to effectively ignore the map η in certain cases.

In order to check that the Radford algebra is a bialgebra, it suffices to check the first two axioms, since the second two are clear from the definition of Δ and ε. We will not do this verification explicitly, but the verification can be seen in the Mathematica code in the appendix.

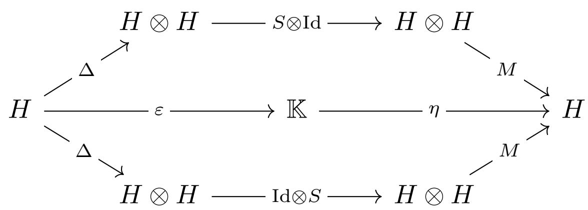
With all the previous definitions done, we are now able to define a Hopf algebra.

Definition 12. A Hopf algebra H is a $\mathbb{K}$-vector space equipped with five $\mathbb{K}$-linear maps

$$\begin{aligned} M &: H \otimes H \to H, \\ \eta &: \mathbb{K} \to H, \\ \Delta &: H \to H \otimes H, \\ \varepsilon &: H \to \mathbb{K}, \\ S &: H \to H, \end{aligned}$$

such that H is a bialgebra under the usual maps, and the following equality holds.

$$S(a_{(1)})a_{(2)} = \varepsilon(a) = a_{(1)}S(a_{(2)}).$$



We should actually write $S(a_{(1)})a_{(2)} = \eta(\varepsilon(a))$, but since H is a bialgebra, we can ignore the η map and interpret the scalar as an element of H.

Remark. It turns out that S is always an antihomomorphism. i.e. $S(ab) = S(b)S(a)$. See for example Radford [10] Proposition 7.1.9 (a).

Now we want to finish the Radford algebra example. We define S on the Radford algebra by $S(g) = g$, $S(x) = -xg = 1 - g + gx$ and $S(y) = -yg = 1 - g + gy$, and extend this to the other elements using the fact that S is an antihomomorphism. This finishes the Radford algebra. Radford discusses this whole algebra the the end of the paper "The Order of the Antipode of a Finite Dimensional Hopf Algebra is Finite" [9].

Example 13. Examples of a class of Hopf algebras are group algebras. Let G be some group and define $\mathbb{K}[G] = \{f : G \to K \mid \text{Only finitely many } g \in G \text{ map to a nonzero element.}\}$. Take the permutation group of five elements S_5 as an example. Then $\mathbb{K}[S_5]$ is a 120 dimensional vector space. We often represent the element $f \in \mathbb{K}[G]$ as $\sum_{g \in G} f(g)g$. For example, the map that sends $g_1 \in G$ to 2, and all other elements in G to zero, will be represented by $2g_1 + \sum_{h \neq g_1} 0h$. We apply the convention that we do not write elements of the form $0h$ in our sum, and also write elements of the form $1g$ as just g. We should check that this is a Hopf algebra by checking what the linear maps are.
Let $f = \sum_{g \in G} f(g)g$ and $k = \sum_{g \in G} k(g)g$, then we define $M(f, k)$ as follows: $M(f, k) = \sum_{h \in G}(\sum_{g \in G} f(g)k(g^{-1}h))h$. Using shorthand notation, we can just write that $M(g, h) = gh$, for all $g, h \in G$, and then extend this linearly. The involution S is defined by $S(rg) = rg^{-1}$ for $r \in \mathbb{K}$ and $g \in G$ and extended linearly. The other maps are $\eta(r) = re$, $\Delta(rg) = r(g \otimes g)$ and $\epsilon(rg) = r$, where e denotes the identity element of G.

5.1 Dual Hopf algebra

If we have a finite dimensional Hopf algebra H, then we can also construct its dual H^*. The dual algebra H^* is defined as the set

$$H^* = \mathrm{Hom}(H, \mathbb{K}),$$

together with dual maps.
If H has basis $\{e_i\}$, then it's often most useful to use the dual basis $\{e^i\}$ of H^* such that $e^i(e_j) = \delta^i_j$, where δ^i_j is the Kronecker delta. We will often write evaluation using brackets. i.e. $\langle \cdot, \cdot \rangle : H \otimes H^* \to \mathbb{K}$, $\langle e_i, e^j \rangle = \delta^j_i$. This bracket notation is bilinear, and allows for more intuitive manipulations of expressions.
We will now give the complete definition of the dual Hopf algebra.

Definition 14. Let H be a finite dimensional Hopf algebra over $\mathbb{K}$. The dual Hopf algebra H^* is also a Hopf algebra over $\mathbb{K}$ defined as $(H^*, M^*, \eta^*, \Delta^*, \varepsilon^*, S^*)$. Here $H^* = \mathrm{Hom}(H, \mathbb{K})$, and the maps are defined as follows.

$$\begin{aligned}
M^*(f,g)(x) &= f(x_{(1)}) \cdot g(x_{(2)}),\\
\eta^*(r)(x) &= (r\varepsilon)(x),\\
\Delta^*(f)(x \otimes y) &= f(xy),\\
\varepsilon^*(f) &= f(\eta(1)),\\
S^*(f)(x) &= f(S(x)).
\end{aligned}$$

Since we said that H is finite dimensional, we can use a basis of H. We can then use the dual basis to get explicit formulas for the dual maps. These formulas are the as follows.

$$\begin{aligned}
M^*(f,g) &= \sum_i f((e_i)_{(1)}) g((e_i)_{(2)}) e^i,\\
\eta^*(r) &= \sum_i (r\varepsilon)(e_i) e^i,\\
\Delta^*(f) &= \sum_{i,j} f(e_i e_j)(e^i \otimes e^j),\\
\varepsilon^*(f) &= f(\eta(1)),\\
S^*(f) &= \sum_i f(S(e_i)) e^i.
\end{aligned}$$

One can check that these maps satisfy all the axioms that make up a Hopf algebra. We will often use the convention to omit the stars on the maps if the distinction is clear from context. We will sometimes also use stars on elements to denote that they are elements in the dual.

Going back to our main example. The Radford algebra also has a dual. Since the Radford algebra is eight dimensional, its dual will also be eight dimensional. It is easiest to define the dual with delta functions. We denote the delta functions with capital letters, and use W for the delta function of 1. We thus get the dual basis $\{W, G, X, Y, GX, GY, XY, GXY\}$. One should be careful though. In the Radford algebra, we have $g \cdot x = gx$, but in this dual algebra, the symbol GX is something that should not be confused with $G \cdot X$. The first one is the delta function on gx, the second one is the multiplication of the delta functions for g and for x. They are different elements; $G \cdot X = 0 \neq GX$. Thus, multiplication in this algebra should always be denoted with a multiplication dot, and cannot be denoted by concatenation. The multiplication table of this algebra can be seen in Table 2. For the full structure read the section about the Radford algebra.

5.2 Pivots, integrals and cointegrals

We have now described what Hopf algebra are, but in what follows we need a Hopf algebras with a few more properties. The main ones being dual-pivotal, unimodular, and counimodular. We start of with the definition of a pivot.

Definition 15. A pivotal Hopf algebra is a Hopf algebra H together with an invertible element $u \in H$ such that $S^2(x) = u^{-1}xu$ for all $x \in H$ and such that $\Delta(u) = u \otimes u$, and $\varepsilon(u) = 1$. Moreover, if the pivot satisfies $u^2 = 1$, then the pivot is said to be involutory.

Definition 16. A finite dimensional Hopf algebra H is called dual-pivotal if H^* is pivotal.

	W	G	X	Y	GX	GY	XY	GXY
W	W	0	X	Y	0	0	XY	0
G	0	G	0	0	GX	GY	0	GXY
X	0	X	0	0	0	XY	0	0
Y	$-XY$	$Y+XY$	0	0	$-XY$	0	0	0
GX	GX	0	0	GXY	0	0	0	0
GY	$GY+GXY$	$-GXY$	$-GXY$	0	0	0	0	0
XY	XY	0	0	0	0	0	0	0
GXY	0	GXY	0	0	0	0	0	0

Table 2: Multiplication table for the dual Radford algebra.

Our main goal is to construct an invariant using Hopf algebras. In [7] the authors found an invariant of closed 3-manifolds by demanding that the antipode S was involutory. We do not want to use this assumption. We do however need some assumption about S^2. This is the main motivation to define a dual-pivotal Hopf algebra. The reason why we introduce involutory dual-pivots is because the proof of the 3-5 move needs it, and because it makes the $\Omega 0$ move involutory.
Next, we discuss unimodular and counimodular. These properties are also needed in the proof of the 3-5 move. We first start of with defining integrals.

Definition 17. A left (or right) integral of H is an element $\mu_L \in H^*$ (or $\mu_R \in H^*$) such that $x_{(1)}\mu_L(x_{(2)}) = \mu_L(x)\eta(1)$ (or $\mu_R(x_{(1)})x_{(2)} = \mu_R(x)\eta(1)$) for all $x \in H$.

Definition 18. A left (or right) cointegral is an element $e_L \in H$ (or $e_R \in H$) such that $x \cdot e_L = \varepsilon(x)e_L$ ($e_R \cdot x = \varepsilon(x)e_R$) for all $x \in H$.

Remark. It is quite easy to confuse the notion for a left integral with that of a right integral, and likewise that of a left cointegral and a right cointegral. Even the notion of cointegral can get confusing as some people write "integral for H^*." For example Radford in [10] says an element $\Lambda \in H$ is a integral for H while Mihalache, Suzuki, Terashima in [7] and Kuperberg in [4] call e_L a left cointegral (of H). Sweedler in [12] also did not use the term cointegral at all and talked about integrals in H and H^*. This means that some extra attention should be given to what an author actually means by integral. We will use Kuperberg's terminology, which corresponds with the above definitions.

We will restate these definitions more graphically in the section on tensor diagrams.

For a finite dimensional Hopf algebra there always exist a nonzero left integral. Moreover, the set of all left integrals forms a one dimensional subspace of H^*. The same is true for right integrals and the respective cointegrals. A proof of the one dimensionality of left integrals is given for the proof for Corollary 5.1.6 in [12]. And a proof for all four statements can be found under Theorem 10.2.2 in [10]. As said before, one needs to be careful about what the author means when they are writing about integrals.

It should be noted that the one dimensional space of left integrals need not be the same as space of right integrals. Similarly, the space of left cointegrals need not be the same as the space of right cointegrals. We get the following definitions for if they are the same.

Definition 19. Let H be a finite dimensional Hopf algebra. We say that H is counimodular if the space of left integrals is equal to the space of right integrals.

Definition 20. Let H be a finite dimensional Hopf algebra. We say that H is unimodular if the space of left cointegrals is equal to the space of right cointegrals.

It should be noted that if H is unimodular and counimodular, then so is H^*, since the two definitions are each other's dual.

We will now give some lemmas and definitions that we will later use during our proofs.
The reason why we give this lemma is because we want to put our assumptions on H, but we will often work with H^*. For now we will define an element that will often come up later.

Definition 21. Let $u \in H^*$ be the pivotal element of finite dimensional H^* and let $\mu_L \in (H^*)^* \cong H$ be a left integral of H^* The element $\omega_L \in H$ is defined as $\langle \omega_L, x\rangle = \langle \mu_L, ux\rangle$, for all $x \in H^*$.

Strictly speaking the above definition uses a lot of abuse of notation, but it will be helpful to view H^* as the main Hopf algebra.

Lemma 22. *If finite dimensional H^* is pivotal with pivot $u \in H^*$, and H is unimodular and counimodular, then $\langle \omega_L, ab\rangle = \langle \omega_L, ba\rangle$, for all $a, b \in H^*$.*

Proof. This result is a consequence of a result found by Radford in [10], theorem 10.5.4 part e. The theorem says that if H is a finite dimensional Hopf algebra with antipode S, and $\lambda \in H^*$ a right integral, and α the H-distinguished grouplike element of H^* that then $\langle S^2(a \leftharpoonup \alpha)b, \lambda\rangle = \langle ba, \lambda\rangle$. We of course want to swap the roles of H and H^*, but we will do that at the end. We thus have for $\mu_R \in H^*$ that

$$\langle S^2(a \leftharpoonup \alpha)b, \mu_R\rangle = \langle ba, \mu_R, \rangle.$$

We have assumed H is unimodular. This means that $\alpha = \varepsilon$, so we get

$$\langle S^2(a)b, \mu_R\rangle = \langle ba, \mu_R\rangle.$$

We now use that S^2 is conjugation by u and $u = u^{-1}$ to get

$$\langle \mu_R, uaub\rangle = \langle \mu_R, ba\rangle.$$

Now we replace b by ub to obtain

$$\langle \mu_R, uab\rangle = \langle \mu_R, uba\rangle.$$

Finally, want to show $\mu_R = \mu_L$. This is true if H is counimodular. We can now interpret H^* to be the original algebra and we are done.

□

Lemma 23. *For any pivotal Hopf algebra we have $S(u) = u^{-1}$, and for a dual-pivotal Hopf algebra we have $S^*(u^*) = (u^*)^{-1}$.*

Proof. We know that u is grouplike, so $\Delta(u) = u \otimes u$. We also know $\varepsilon(u) = 1$. The Hopf axiom for S then tells us that $u \cdot S(u) = 1$. In other words, $S(u) = u^{-1}$. The proof for H^* is similar. □

Corollary 24. *For a pivotal Hopf algebra with involutory pivot, we have $S(u) = u$. Similarly for a dual-pivotal Hopf algebra with involutory dual-pivot, we have $S^*(u^*) = u^*$.*

Lemma 25. *We have $\omega_L(a) = \omega_L(S^*(a)), \quad \forall a \in H^*$.*

Proof. This is a consequence of H being unimodular. This means

$$\begin{aligned}
\omega_L(S^*(a)) &= \mu_L(uS^*(a)) \\
&= \mu_L(S^*(u)S^*(a)) \\
&= \mu_L(S^*(au)) \\
&= \mu_L(S^*((S^*)^2(au))) \\
&= \mu_L((S^*)^3(ua)) \\
&= \mu_L(ua) \\
&= \omega_L(a).
\end{aligned}$$

Here we have used $\mu_L(S^*(a)) = \mu_R(a)$. and $\mu_R(S^*(a)) = \mu_L(a)$. Both equalities are only simultaneously true if H is unimodular. □

This lemma also immediately implies $\omega_L(a) = \omega_L((S^*)^3(a))$ by just repeated application of S^* on the input.

Lemma 26. *We have $\omega_L(e_R) = 1$.*

Proof. In Kupperberg's paper Non-Involutory Hopf Algebras and 3-Manifold Invariants [4] he writes in just before Lemma 3.6 on page 24 that $\mu_L(e_R) = 1$. We have $\omega_L(e_R) = \mu_L(ue_R)$. Using that e_R is a right cointegral in H^*, we get $\omega_L(e_R) = \mu_L(e_R)\varepsilon(u) = 1$. □

5.3 The Heisenberg Double

We will also work a lot with the Heisenberg double. This is an algebra created out of a Hopf algebra via a certain construction. The Heisenberg Double is not a Hopf algebra.

Definition 27. Let H be a Hopf algebra. The Heisenberg double of H is defined as

$$\mathcal{H}(H) = H \otimes H^*.$$

The multiplication on this space is defined as.

$$(a \otimes f)(b \otimes g) = \langle b_{(2)}, f_{(1)} \rangle a \cdot b_{(1)} \otimes f_{(2)} \cdot g =: ab_{(1)} \otimes (b_{(2)} \leftharpoonup f)g$$

We use notation $(a \rightharpoonup f) = \langle a, f_{(2)} \rangle f_{(1)}$ and $(a \leftharpoonup f) = \langle a, f_{(1)} \rangle f_{(2)}$. These are maps can be interpret as follows: $(a \rightharpoonup f)(x) = f(xa)$ and $(a \leftharpoonup f)(x) = f(ax)$.
In the proof of the invariance under the MP moves we will show where this multiplication arises.

In $\mathcal{H}(H) \otimes \mathcal{H}(H)$ is a special element called the universal R-matrix. We will denote this matrix by T. Since it is an element of $\mathcal{H}(H) \otimes \mathcal{H}(H)$, it has two tensor components. Using Sweedler notation, we will denote the first component by T_1 and the second component by T_2. In other words $T = T_1 \otimes T_2$. The R-matrix also has a inverse denoted by T^{-1}. This too can be written as $T^{-1} = T_1^{-1} \otimes T_2^{-1}$. Lastly, there is a third elements we call $\overline{T}$. This is can also be written as $\overline{T} = \overline{T}_1 \otimes \overline{T}_2$. We will also often use shorthand notation by which we write elements like $a \otimes \varepsilon \in \mathcal{H}(H)$ as just $a \in \mathcal{H}(H)$, and $1 \otimes f$ as just f. Using shorthand notation, the formulas for T, T^{-1} and $\overline{T}$ are

$$\begin{aligned} T &= \sum_i e_i \otimes e^i \in \mathcal{H}(H)^{\otimes 2}, \\ T^{-1} &= \sum_i S(e_i) \otimes e^i \in \mathcal{H}(H)^{\otimes 2}, \\ \overline{T} &= \sum_i S^{-1}(e_i) \otimes e^i \in \mathcal{H}(H)^{\otimes 2}. \end{aligned}$$

5.4 Assumptions on H and H^*

In order for our invariant to work, we need to assume a few things about H. We first assume H is finite dimensional. Firstly because we can easily find the R-matrix T for finite dimensional H. Secondly, existence of integrals and cointegrals is also guaranteed for finite dimensional Hopf algebras. Lastly, because the notion of a dual Hopf algebra is is easily obtained in the finite dimensional case.

Our second assumption is that H is unimodular and counimodular. Counimodular means that that H^* is unimodular, which allows $\omega_L(ab) = \omega_L(ba)$. We need H to be counimodular, because we need $\mu_L = \mu_R$, where $\mu_L \in H$ is a cointegral of H, which is an integral in H^*. By a result of Radford, a Hopf algebra that is unimodular and counimodular satisfies $S^4 = \mathrm{Id}$.

Lastly we want that H is dual-pivotal. So H^* has a pivotal element. i.e. an element $u \in H^*$ such that $(S^*)^{-2}(x) = u^{-1}xu$. We also want that u is involutory, i.e. $u^{-1} = u$. Being dual-pivotal with involutory dual-pivot also implies $(S^*)^4 = \mathrm{Id}$.

5.5 The Radford Algebra

In Radford's paper "The Order of the Antipode of a Finite Dimensional Hopf Algebra is Finite" [9] he gives a whole family of Hopf algebras, and ends his paper with a slight modification of one of his examples. This algebra, which we from here on out will call the Radford algebra, is constructed as follows. Our notation will be slightly different from his.

Consider the free algebra over $\mathbb{K}$ generated by the following non-commuting indeterminants g, x and y. It has a bialgebra structure given by

$$\begin{aligned} \Delta(g) &= g \otimes g, \\ \Delta(x) &= x \otimes g + 1 \otimes x, \\ \Delta(y) &= y \otimes g + 1 \otimes y, \\ \varepsilon(g) &= 1, \\ \varepsilon(x) &= \varepsilon(y) = 0. \end{aligned}$$

We quotient this algebra by de ideal generated by $\{g^2-1, x^2-x, y^2-y, gx+xg-(g-1), gy+yg-(g-1), xy+yx-(x+y)\}$.
The antipode maps are generated by

$$\begin{aligned} S(g) &= g, \\ S(x) &= 1-g+gx, \\ S(y) &= 1-g+gy. \end{aligned}$$

We then end up with a unimodular, counimodular, pivotal, dual-pivotal Hopf algebra. For the dual, we will use capital letters for the basis $\{W, G, X, Y, GX, GY, XY, GXY\}$. Each capital letter is the deltafuncion which yields 1 on the corresponding lower case letter, and where W corresponds to $W(1) = 1$. As said before, multiplication in the dual cannot be denoted by concatenation. A normalized integral of H is $\Lambda = XY$ and a cointegral is $\lambda = xy - x + gxy - gx$. The Radford algebra has a pivot, namely $u = g$ and the dual-pivot is $u^* = W + X + Y + XY - G - GX - GY - GXY$. These element satisfy $u^2 = 1$, $\Delta(u) = u \otimes u$ and $\varepsilon(u) = 1$ in H, and $(u^*)^2 = 1$, $\Delta^*(u^*) = u^* \otimes u^*$, and $\varepsilon^*(u^*) = 1$ in H^*. For comultiplication structure of H see the code in the appendix.

Lastly, the unit map of $\mathcal{R}^*$ is given by $\eta^*(1) = W + G$. The counit on the basis is given by

$$\varepsilon^*(x) = \begin{cases} 1 & x = W, \\ 0 & \text{else}, \end{cases}$$

and extended linearly. The antipode is given by

$$\begin{aligned} S^*(W) &= W + X + Y + XY, \\ S^*(G) &= G - X - Y - XY, \\ S^*(X) &= -GX - XY - GXY, \\ S^*(Y) &= -GY, \\ S^*(GX) &= X, \\ S^*(GY) &= Y + XY - GXY, \\ S^*(XY) &= XY, \\ S^*(GXY) &= GXY. \end{aligned}$$

6 Construction of the invariant

In this section we describe the process to go from a spin o-graph to an invariant.

Recall that a spin o-graph carries the following information. Firstly, it carries a direction on the edges. Secondly, at each vertex it carries a notion of edges going over and under each other. Lastly, there are is the color of an edge. The first two pieces of information allow us to make a distinction between so called positive and negative crossings, as seen in Figure 14.

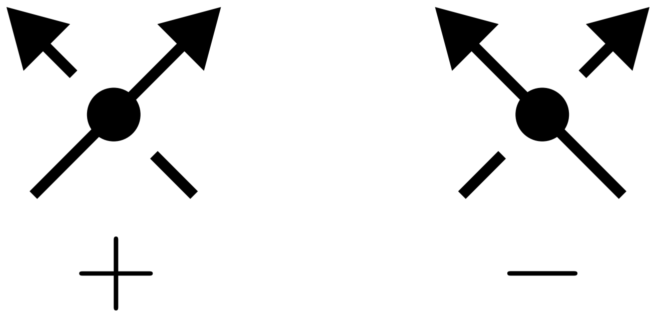


Figure 14: A positive (on the left) and negative (on the right) crossing.

We will make our o-graph into a $\mathcal{H}(H)$ decorated diagram. This means that we will turn our o-graph into a closed curve on which we will place dots that represent elements of $\mathcal{H}(H)$. Recall that we have the elements $T, T^{-1} \in \mathcal{H}(H)^{\otimes 2}$. We will put the elements T and T^{-1} on the vertices in the way depicted in Figure 15. The choice to put either T or T^{-1} on the strand depends on wether the crossing is positive or negative. Of course, T is not an element of $\mathcal{H}(H)$, but of $\mathcal{H}(H)^{\otimes 2}$. Recall that T can be written as $T = T_1 \otimes T_2$, with $T_1, T_2 \in \mathcal{H}(H)$. These last two elements are the elements we actually put on the strands. The first tensor factor always goes on the under strand, and the second tensor factor on the over strand.

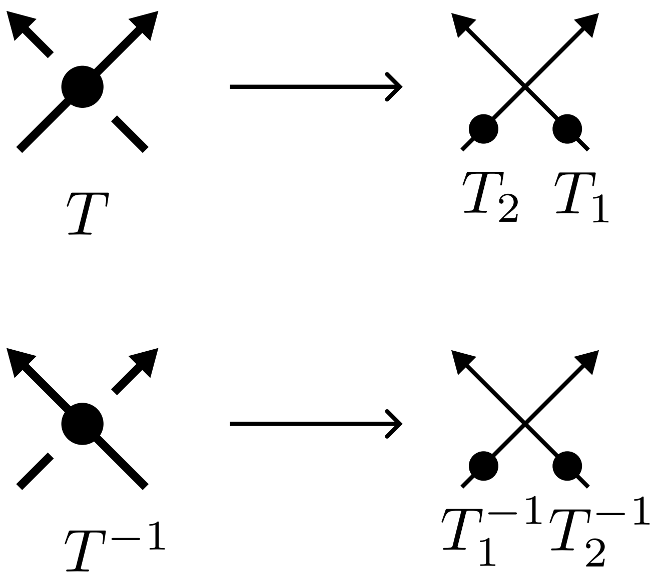


Figure 15: Form a vertex in an o-graph to decorated diagram.

We also want to put elements on the strands that have a nonzero spin number. We have assumed there is an involutory pivotal element of H^*, namely $u \in H^*$. We extend this to $1 \otimes u \in \mathcal{H}(H)$. It is this element that we put on a strand with a spin number.

When we have our decorated diagram, there are two steps we need to do that can be done in any order. We either multiply the elements together in the Heisenberg double, and take the representation, or we take the representation of all the elements and matrix multiply them together. We will often use the former method. For now we will discuss the representation.

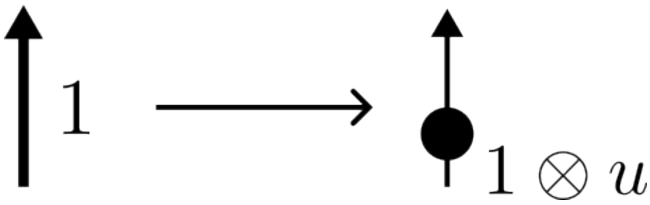


Figure 16: From a edge with spin number to decorated diagram.

6.1 Fock representation

The representation we will use is the Fock representation. This is a map $\rho : \mathcal{H}(H) \to \mathrm{End}(H^*)$. It defined as follows:

$$\rho : \mathcal{H}(H) \to \mathrm{End}(H^*)$$
$$\rho(a \otimes f)(g) = \langle a, g_{(1)} \rangle g_{(2)} \cdot f.$$

Lemma 28. *We have $\rho(a \otimes f)(g)(x) = g(ax)f(x)$.*

Proof. The Fock representation yield a product of a scalar and two functions. We know how evaluation of a product is defined, namely as follows $\rho(a \otimes f)(g)(x) = \langle x, \langle a, g_{(1)} \rangle g_{(2)} \cdot f \rangle$. If we pull out the inner bracket we get $\langle a, g_{(1)} \rangle \langle x, g_{(2)} \cdot f \rangle$. Using that multiplication is dual to comultiplication we get $\langle a, g_{(1)} \rangle \langle x_{(1)}, g_{(2)} \rangle \langle x_{(2)}, f \rangle$. We can combine the first two brackets to get $\langle ax_{(1)}, g \rangle \langle x_{(2)}, f \rangle$, which is the definition of $g(ax)f(x)$, hence $\rho(a \otimes f)(g)(x) = g(ax)f(x)$. □

The above lemma is not really needed for anything technical, but it gives a good intuition on what the Fock representation does.

This representation can take us from a $\mathcal{H}(H)$-decorated diagram to a tensor diagram in H^*. This conversion will be shown for the elements T and T^{-1} in the chapter about representations. We will now briefly explain what a tensor diagram is.

6.2 Tensor diagram

We will graphically depict elements of $\mathrm{End}(H^*)$ by tensor diagrams. A tensor diagram is a way to describe what happens to arbitrary vectors, matrices and higher order tensors by means of writing down what elements affect others. More concretely, given a vector space V, we will denote a rank (n, m) tensor $A \in V^{\otimes n} \otimes (V^*)^{\otimes m}$ as with a diagram of m arrows going into A and n arrows leaving A, as seen in Figure 17. The general convention is that the arrows are counted in the following way. The ingoing arrows are counted from the top of the element going counterclockwise, and the outgoing arrows are counted from the top going clockwise.
We will often encounter elements that have either two ingoing arrows and one outgoing, or one ingoing arrow and two outgoing. In this case we use the following convention. In the case of two incoming arrows and one outgoing, we count from the outgoing arrow going counterclockwise. In the case of two outgoing arrows and one incoming, we count from the incoming arrow going clockwise. This makes it so that these specific types of tensor diagrams do not change based on the rotation of the diagram, whereas the normal convention does care about that.

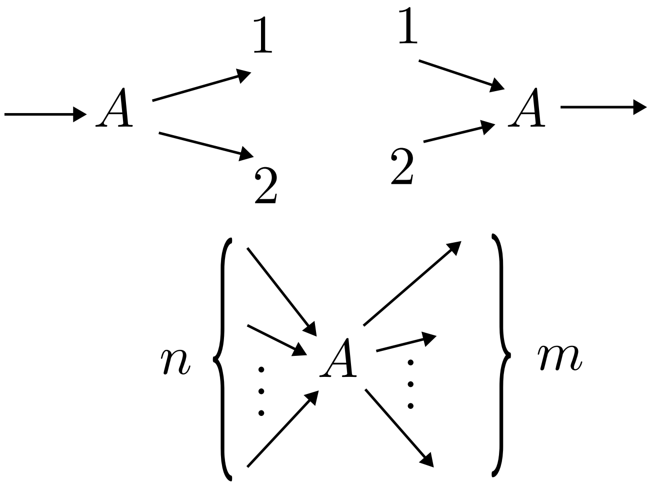


Figure 17: Tensor diagrams. The top two indicate our often used convention.

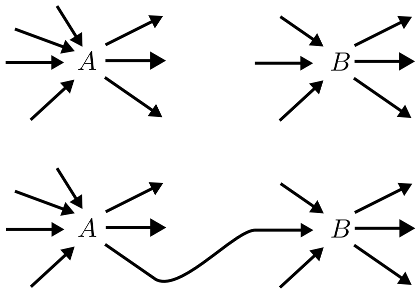


Figure 18: Contracting tensors A and B.

We denote two tensors multiplying together by joining their arrows. For example, a vector is an element with one outgoing arrow. When we multiply a vector by matrix, we get another vector. This means that a matrix is a tensor element with one ingoing and one outgoing arrow. Similarly, the multiplication map M of a Hopf algebra has two incoming arrows and one outgoing. These incoming arrows represent the two elements that are being multiplied, and the outgoing arrow is the result. The comultiplication map Δ has one incoming arrow and two outgoing arrows. The incoming arrow represents the element that is being comultiplied, and the two outgoing arrows are the two tensor factors. An example of tensor multiplication is given in Figure 18. Here we see that A is a $(3, 4)$ tensor and B is a $(3, 3)$ tensor.The third outgoing arrow of A is contracted by the second ingoing arrow in B.

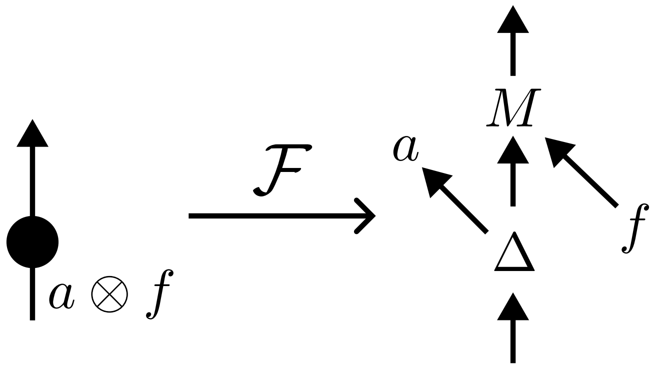


Figure 19: The Fock representation of $a \otimes f$ as a tensor diagram.

We will make heavy use of diagrams in our proof of the invariance under the Pontrjagin move, but we have not yet explicitly explained what the tensor diagram of a positive or negative vertex should be. It drops out almost immediately from the definitions of T, T^{-1} and the Fock representation. Recall that the Fock representation was given by

$$\rho : \mathcal{H}(H) \to \mathrm{End}(H^*)$$
$$\rho(a \otimes f)(g) = \langle a, g_{(1)} \rangle g_{(2)} \cdot f,$$

and that $T = \sum_i e_i \otimes e^i = T_1 \otimes T_2$. We had that we put T_1 on the right understand of a positive crossing, and T_2 on the left over stand. If we read Figure 19, we see that we end up with Figures 20 and 21.

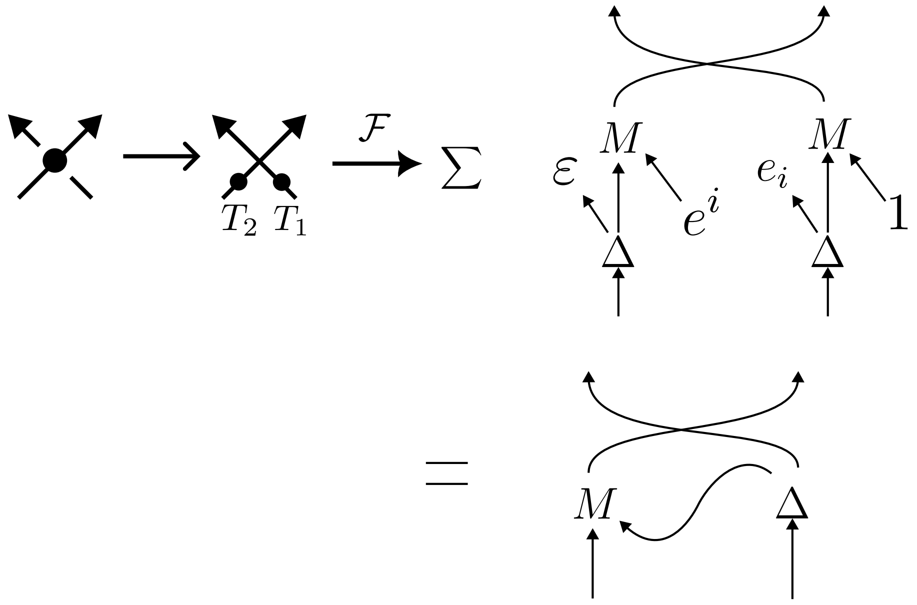


Figure 20: The tensor diagram for T.

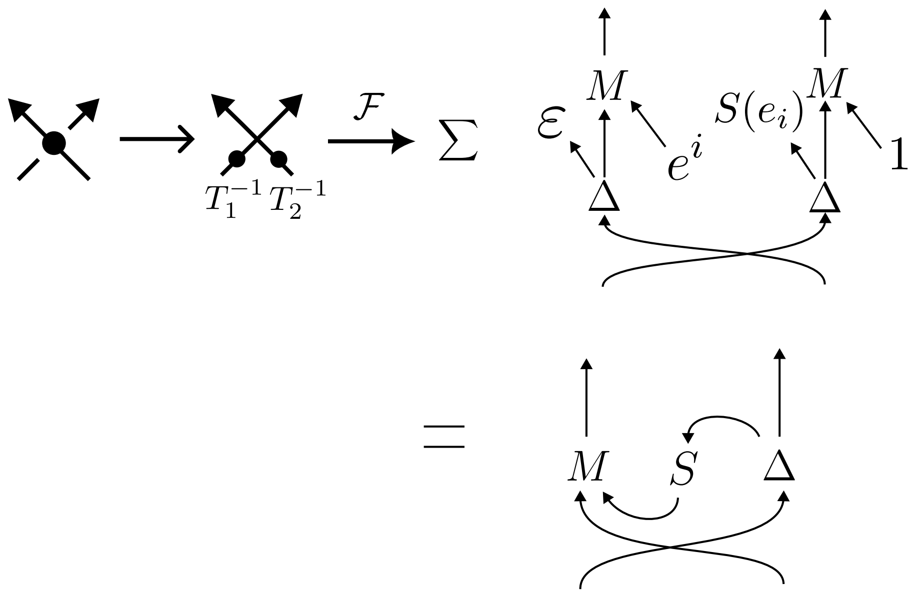


Figure 21: The tensor diagram for T^{-1}.

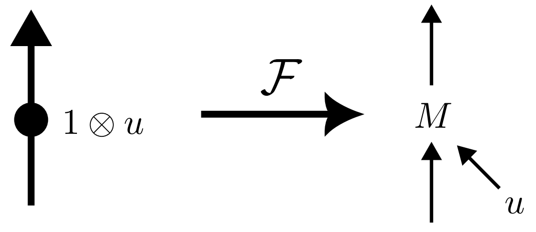


Figure 22: The tensor diagram for $1 \otimes u \in \mathcal{H}(H)$.

It should be noted the we have abused notation and that all the maps should have a star on it. But this leads to too much clutter so we will ignore the stars on the maps and write them as is.

We also want to visually depict the definitions we gave in Section 5.2. The definitions of a cointegral and integral can be seen in Figure 23.

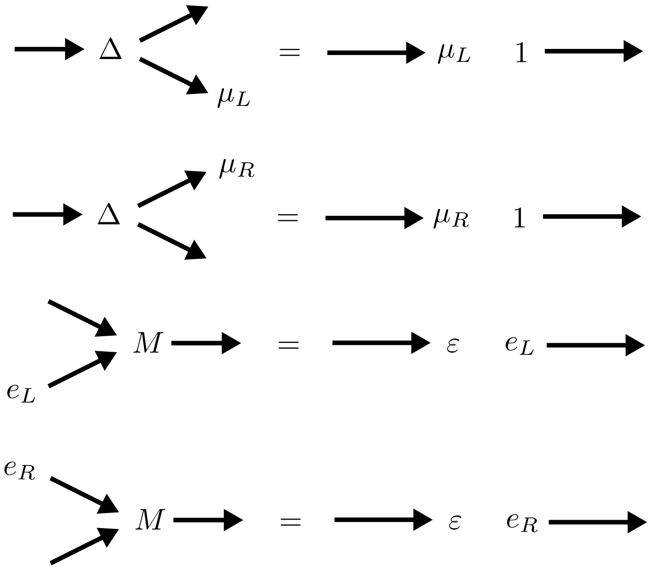


Figure 23: The definitions of cointegrals and integrals as tensor diagrams.

6.3 Continuation of construction

Now that we have discussed the Fock representation, we will continue with the construction. Like previously said, one can either multiply all elements in the decorated diagram first and then use the Fock representation on the result, or take the Fock representation and then matrix multiply all the results. As said before we will mainly look at the the first option. We have not said where one should start with multiplying. Since multiplication in a Hopf algebra need not be commutative, the result of multiplying all the elements along the decorated diagram is dependent on a starting point. This is one of the main reasons why we take the Fock representation. We pick any arbitrary starting point and multiply the elements together along the strand. We apply the Fock representation to get a matrix in $\mathrm{End}(H^*)$. We then take the trace of the matrix. This is the most important step. Since the trace satisfies $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ for any matrix A and B, we get that the result is a scalar in the base field $\mathbb{K}$ of the Hopf algebra H, which is independent of the chosen stating point. Of course, one can also first take the Fock representation of each element, matrix multiply them together by choosing an arbitrary starting point to multiply from, and then take the trace. The result is the same.

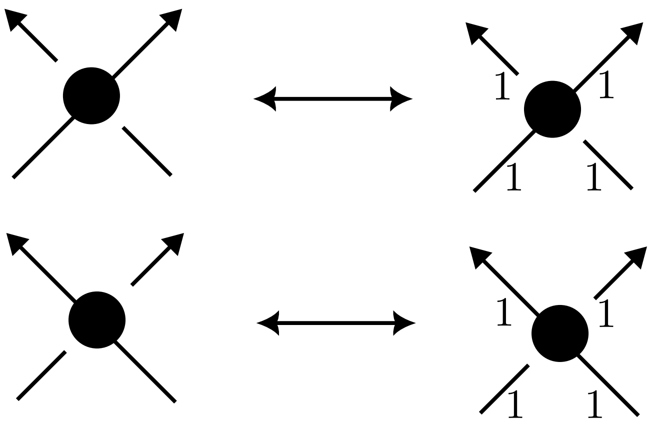


Figure 24: The Ω0

The obtained number is not yet the invariant of our spin manifold. We still have one complication. This procedure is not invariant under the Ω0 move. Luckily there is a remedy for this which means that we have found an almost invariant. We define an almost invariant in the following way.

Definition 29. An almost invariant of a framed / spin o-graph is a function that is invariant under the 0-2 move and MP move for framed manifolds, and in the case of spin manifolds also the 3-5 move. It should also make the Ω0 move involutory.

An almost invariant is a true/usual invariant as in the normal use of the word. It is just a specific type of invariant. It will not be just an element of the ground field $\mathbb{K}$ of the Hopf algebra H, but a finite set of elements in the ground field; $\cup_{i\in I}\{r_i\}$, $r_i \in \mathbb{K}$, and $|I| < \infty$.

At this stage, we do not have invariance under the $\Omega 0$ move, while we do need that for a invariant. Like we said, we want $\Omega 0$ to be involutory for our invariant to work. With that assumption we can use a trick to get around the problem of not being invariant under $\Omega 0$. To see the trick we first need to define the complexity of a manifold.

Definition 30. Let M be a framed/ spin manifold with framed / spin o-graph Γ. The complexity of M is the minimum amount of vertices that Γ can can have. In other words, it is the amount of vertices in the reduced o-graph of M. The complexity of M will be denoted by $c(M)$.

For a reference on complexity, see Matveev's book [5] Chapter 2.

Now the final step of our invariant. If the spin o-graph that we discuss has k true vertices, then we compute the scalar that we obtain from all 2^k different o-graphs that are obtained by either applying $\Omega 0$ to a vertex, or not applying $\Omega 0$. The set of all these scalars will be our invariant, which we will call $\mathcal{Z}(M, s; H)$, where M is the oriented, compact, connected 3-manifold with empty boundary, s is the spin structure on M, and H is the finite dimensional, dual-pivotal, unimodular, counimodular Hopf algebra. It is a finite set whose elements are elements from $\mathbb{K}$, and which cardinality is at most $2^{c(M)}$. We are not entirely sure if this method is truly invariant, but there is a (non-rigorous) argument to be made, namely the following.

If you apply all $\Omega 0$ moves to the o-graph that you have, then you have mapped out all possible ways $\Omega 0$ can affect the particular o-graph that you have. Note that if you apply $\Omega 0$ to a particular vertex twice, you essentially have not changed anything. If you then apply a 0-2 move, an MP move, or a 3-5 move, then the invariant number you get is still the same as before. The number of vertices however might have changed. We reckon, without proof, that applying a $\Omega 0$ move to this new graph would yield a number already obtained by some other permutation of applying $\Omega 0$ to the original. This means that exploring an o-graph by $\Omega 0$ in the state that it is in, is the same as exploring the o-graph with the amount of vertices being minimal, hence you always get a set that has cardinality less than $2^{c(M)}$.

6.4 Examples of the invariant

(a) A spin o-graph of $\mathbb{S}^3$. (b) A spin o-graph of $\mathbb{PR}^3$.

Figure 25: Two examples of spin o-graphs

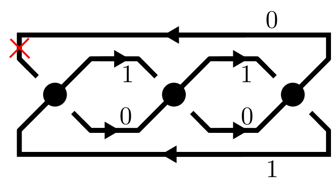


Figure 26: A spin o-graph of $\mathbb{S}^3/Q_8$.

Take a lens space $L(2,1)$ with Radford algebra. We will use the spin structure on the Lens space given by [6], which corresponds to the spin o-graph of Figure 5. Call this particular spin Lens o-graph $\Gamma_{L(2,1)}$.

We have four different diagrams that we have to consider. Starting from the left side including the 1 (start at the red cross) we get the four elements

$$uT_2uT_2'uT_1'T_1,$$
$$uT_2T_2'uT_1'uT_1,$$
$$uT_2T_2'uT_1'uT_1,$$
$$uT_2uT_2'uT_1'T_1.$$

Here we have used primes to to differentiate between the left and the right crossing. They are the same elements, but it is purely for visual clarity. We now calculate

$$\mathrm{Tr}(\rho(uT_2uT_2'uT_1'T_1)) = 4,$$
$$\mathrm{Tr}(\rho(uT_2T_2'uT_1'uT_1)) = 4.$$

This means that $\mathcal{Z}(\Gamma_{L(2,1)}; s_\Gamma; \mathcal{R}) = \{4\}$, to which we remark that the size of the set is not strange, as Matveev has written in [5] that the complexity of $\mathbb{RP}^3$ is equal to 0. Here f_Γ denotes the spin structure of the manifold, since the result of the construction depends on it.
What is also remarkable is that this does correspond with the findings of [6], where they conclude that the invariant is $\mu_L(e_R) \cdot \mathrm{Tr}(S)$, which for us would yield 4.

We can also look at a spin 3-sphere with a spine structure as in Figure 4. We find

$$\mathrm{Tr}(\rho(uT_1T_2)) = 1.$$

Since the $\Omega 0$ move on this o-graph does nothing, we get that our invariant is $\{1\}$.

Lastly for $\mathbb{S}^3/Q_8$. Matveev says this has complexity two, so we should expect at most four different results if we compute the trace of the Fock representation. The elements that we should take the Fock representation off are:

$$T_1^{-1}T_2'^{-1}uT_1''^{-1}uT_2^{-1}uT_1'^{-1}T_2''^{-1},$$
$$T_1^{-1}uT_2'^{-1}uT_1''^{-1}T_2^{-1}T_1'^{-1}T_2''^{-1}u,$$
$$T_1^{-1}uT_2'^{-1}T_1''^{-1}uT_2^{-1}T_1'^{-1}uT_2''^{-1},$$
$$T_1^{-1}T_2'^{-1}T_1''^{-1}T_2^{-1}uT_1'^{-1}uT_2''^{-1}u,$$
$$T_1^{-1}T_2'^{-1}T_1''^{-1}T_2^{-1}uT_1'^{-1}uT_2''^{-1}u,$$
$$T_1^{-1}uT_2'^{-1}T_1''^{-1}uT_2^{-1}T_1'^{-1}uT_2''^{-1},$$
$$T_1^{-1}uT_2'^{-1}T_1''^{-1}uT_2^{-1}T_1'^{-1}T_2''^{-1}u,$$
$$T_1^{-1}T_2'^{-1}uT_1''^{-1}uT_2^{-1}uT_1'^{-1}T_2''^{-1}.$$

Note the first four equations are the same as the last four. This means that the size of the invariant set will indeed be less then or equal to four. We want to calculate $\mathrm{Tr}(\rho(T_1^{-1}T_2'^{-1}uT_1''^{-1}uT_2^{-1}uT_1'^{-1}T_2''^{-1})$ using our program, but sadly it is too unoptimized to carry out this calculation within reasonable time, let alone all four of them.

7 Proof of invariance

In this section we will provide a proof of the invariance under the 0-2 move, the MP moves and, the 3-5 move. We do not provide a proof of invariance under the $\Omega 0$ move, as our construction is not invariant under it. Instead, one should read the informal argument we gave at the end of Section 6.3. Our proofs will also highlight how certain constructions came to exist. The best order to read them in, in order to see the construction, would be first the proof of the MP moves, then the 0-2 move, and then the 3-5 move.

7.1 The 0-2 move

The first move of combing calculus, also called the 0-2 move can be seen in Figure 6 or below. This is the move as discussed in [2].

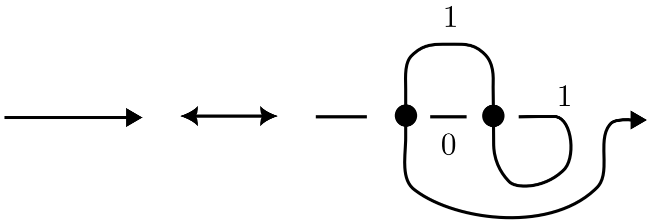


Figure 27: The 0-2 move again.

We will not look at the move in this form. Instead, we will cut the move along these red lines as seen in Figure 28.

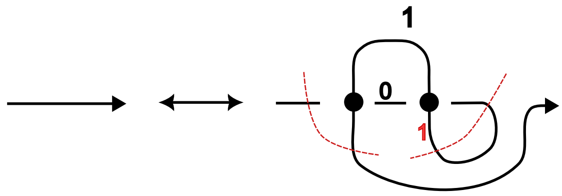


Figure 28: We cut the 0-2 move.

Note that we have moved the red 1 to a specific spot. Having done this, the local picture of this move now becomes Figure 29.
We can see $u \in H^*$ as an element in $\mathcal{H}(H)$, namely the element $1 \otimes u \in \mathcal{H}(H) = H \otimes H^*$. Using shorthand notation we write just $u \in \mathcal{H}(H)$. We define $\bar{T} \in \mathcal{H}(H)^{\otimes 2}$ as $\bar{T} := (1_{\mathcal{H}} \otimes u)T^{-1}(1_{\mathcal{H}} \otimes u) = \sum_i S^{-1}(e_i) \otimes e^i$. We will show the second equality later. We want to show that the following equation holds:

$$T_1\bar{T}_1\bar{T}_2T_2 = 1 \otimes \varepsilon = 1_{\mathcal{H}(H)}.$$

In the above, $T = T_1 \otimes T_2 \in \mathcal{H}(H)^{\otimes 2}$ and $\bar{T} = \bar{T}_1 \otimes \bar{T}_2 \in \mathcal{H}(H)^{\otimes 2}$.
Instead of the above, we will show a different equality, namely the one in Figure 29. The right hand side of the Figure is equal to the following:

$$\sum_{i,j} e_i S^{-1}(e_j) \otimes e^j e^i.$$

We recall the antipode axiom $x_{(1)}S(x_{(2)}) = \varepsilon(x)$. We know S is a antihomomorphism. This means that when we apply S^{-1} to both sides, we get

$$x_{(2)}S^{-1}(x_{(1)}) = S^{-1}(\varepsilon(x)) = \varepsilon(x).$$

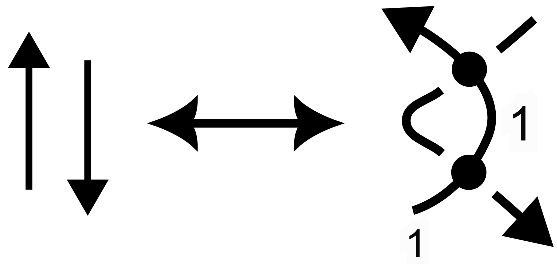


Figure 29: Solving this is equivalent to solving 0-2.

We are able to compare coefficients with the original expression. If we compare with $x \in H$ we obtain

$$\begin{aligned}\sum_{i,j} \langle x, e^j e^i \rangle e_i S^{-1}(e_j) &= \sum_{i,j} \langle x_{(1)}, e^j \rangle \langle x_{(2)}, e^i \rangle e_i S^{-1}(e_j) \\ &= \sum_j x_{(2)} \langle x_1, e^j \rangle S^{-1}(e_j) \\ &= \sum_j x_{(2)} S^{-1}(\langle x_1, e^j \rangle e_j) \\ &= x_{(2)} S^{-1}(x_{(1)}) \\ &= \varepsilon(x) \\ &= \varepsilon(x) \cdot 1.\end{aligned}$$

We see that for all $x \in H$ that $T_1 \bar{T}_1 \langle x, \bar{T}_2 T_2 \rangle = \varepsilon(x)1$. Since it is true for all $x \in H$, it follows that our expression is indeed equal to $1 \otimes \varepsilon$.

Finally, we want to show that $(1 \otimes u)T^{-1}(1 \otimes u) = \bar{T}$. We have

$$\begin{aligned}(1 \otimes u)T^{-1}(1 \otimes u) &= \sum_i S(e_i) \otimes u e^i u \\ &= \sum_i S(e_i) \otimes (S^*)^2(e^i).\end{aligned}$$

If we again do the same trick of comparing $\bar{T}$ with $x \in H$ we get the following:

$$\begin{aligned}&\sum_i \langle x, e^i \rangle S^{-1}(e_i) \\ &= S^{-1}(x).\end{aligned}$$

We now want to compare our expression involving u with x. We get

$$\begin{aligned}\sum_i \langle x, (S^*)^{-2}(e^i) \rangle S(e_i) &= \sum \langle S^{-2}(x), e^i \rangle \rangle S(e_i) \\ &= \sum_i S(\langle S^{-2}(x), e^i \rangle e_i) \\ &= S(S^{-2}(x)) \\ &= S^{-1}(x).\end{aligned}$$

This proves $(1 \otimes u)T^{-1}(1 \otimes u) = \bar{T}$.

7.2 The MP moves

In this section, we will show how the multiplication in the Heisenberg double arises from a few simple assumptions and a specific MP move; MP 4.2. We want to show invariance under the MP moves. MP 4.2 is shown in Figure 30.

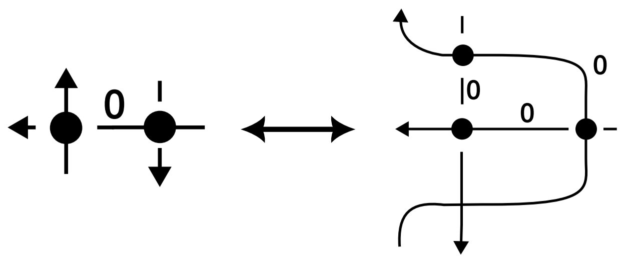


Figure 30: MP 4.2

The MP 4.2 move is easiest to work with. There are no rotation numbers in it, and all crossings are positive. The last special thing about this move is that the right hand side is in so called UO-form. If we follow a strand, it will first do all crossings under, and then all crossings over. The left hand side has a single strand that is in OU-form. We drop our shorthand notation and will write an element in $\mathcal{H}(H)$ out fully as a tensor product for now. Suppose we do not know what the multiplication in $H \otimes H^*$ should be. It can be derived in the following way. We start with three assumptions about how multiplication should behave, namely that

$$(a \otimes \varepsilon)(b \otimes \varepsilon) = (ab \otimes \varepsilon)$$
$$(1 \otimes f)(1 \otimes g) = (1 \otimes fg)$$
$$(a \otimes \varepsilon)(1 \otimes g) = (a \otimes g)$$

for all $a, b \in H$ and $f, g \in H^*$. We should now be able to define $(1 \otimes f)(b \otimes \varepsilon)$ using the above assumptions and the MP 4.2 move. We will now continue again using the shorthand notation.

We let the right vertical strand correspond to the first tensor factor, the left vertical strand to the second factor, and the horizontal strand to the third. The left crossing in the left hand side will correspond to the index i, and the right crossing to index j. For the right side of the diagram, we will name the crossing i, j and k, starting from the top crossing and going counterclockwise. We get the equation

$$\sum_{i,j} e_j \otimes e^i \otimes e^j e_i = \sum_{i,j,k} e_k e_j \otimes e^i e^k \otimes e_i e^j.$$

We have the map $\langle \cdot, \cdot \rangle : H \otimes H^* \to \mathbb{K}$ by $\langle e_i, e^j \rangle = \delta_i^j$ and extend that linearly. We know that for any $x \in H$ that

$$x = \sum_k \langle x, e^k \rangle e_k,$$

and similarly $y = \sum_k \langle e_k, y \rangle e^k$.

We have to match the coefficients of $e^j e_i$ with those on the right hand side. We apply $\langle \cdot, \cdot \rangle$ to the first and second tensor factor such that they disappear on the left hand side. After renaming some of the indices we get

$$e^j e_i = \sum_{k,l,m} \langle e_i, e^k e^m \rangle \langle e_m e_l, e^j \rangle e_k e^l.$$

We can use the fact that comultiplication is the dual to normal multiplication to get

$$\begin{aligned} e^j e_i &= \sum_{k,l,m} \langle e_i, e^k e^m \rangle \langle e_m e_l, e^j \rangle e_k e^l \\ &= \sum_{k,l,m} \langle \Delta(e_i)_{(1)}, e^k \rangle \langle \Delta(e_i)_{(2)}, e^m \rangle \langle e_m, \Delta(e^j)_{(1)} \rangle \langle e_l, \Delta(e^j)_{(2)} \rangle e_k e^l \\ &= \sum_m \langle \Delta(e_i)_{(2)}, e^m \rangle \langle e_m, \Delta(e^j)_{(1)} \rangle \Delta(e_i)_{(1)} \Delta(e^j)_{(2)}. \end{aligned}$$

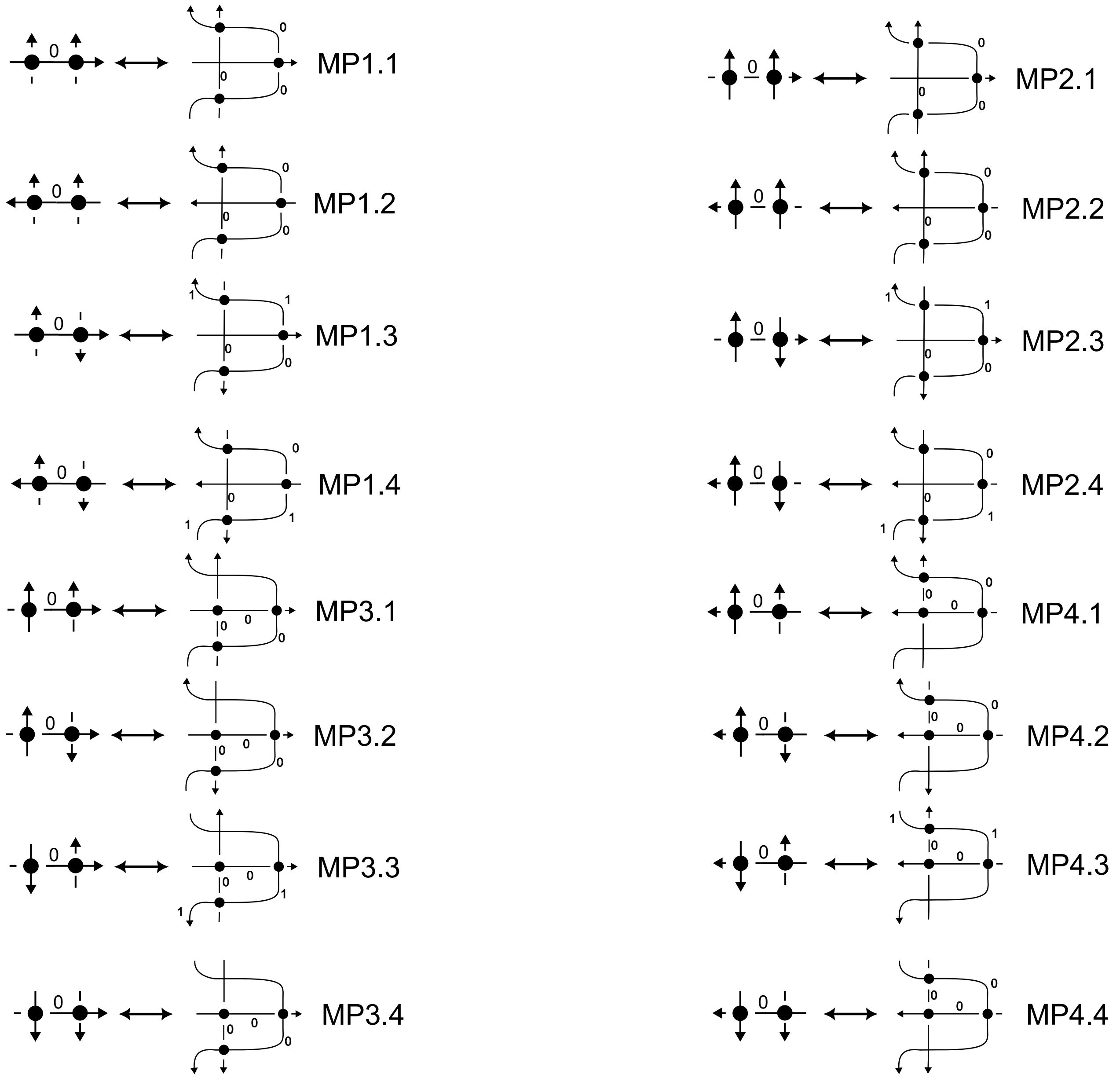
There is one final step we can do to simplify this expression. Observe that $\sum_m \langle \Delta(e_i)_{(2)}, e^m \rangle$ is a constant. It can thus be moved inside the other brackets. We obtain

$$\begin{aligned} e^j e_i &= \sum_m \langle \Delta(e_i)_{(2)}, e^m \rangle \langle e_m, \Delta(e^j)_{(1)} \rangle \Delta(e_i)_{(1)} \Delta(e^j)_{(2)} \\ &= \left\langle \sum_m \langle \Delta(e_i)_{(2)}, e^m \rangle e_m, \Delta(e^j)_{(1)} \right\rangle \Delta(e_i)_{(1)} \Delta(e^j)_{(2)} \\ &= \langle \Delta(e_i)_{(2)}, \Delta(e^j)_{(1)} \rangle \Delta(e_i)_{(1)} \Delta(e^j)_{(2)}. \end{aligned}$$

The main issue is that MP-move 4.2 is not the only move; there are 15 more moves. Let us take for example move 1.1. The way to transform this equation to 4.2 is as follows. First we multiply from the right by T_{21}. This can be visualized by adding an extra crossing at the end. Then we change the names of strand 1 and 2 by using $\tau : \mathcal{H}(H)^{\otimes 2} \to \mathcal{H}(H)^{\otimes 2}$ by $\tau(x \otimes y) = y \otimes x$.
We can transform any MP move to MP 4.2 only by multiplying with T, T^{-1}, $\bar{T}$, swapping strands, and the use of the $\Omega 0$ move.
Recall that $T \cdot T^{-1} = 1 \otimes 1$ and $(u \otimes 1)T^{-1}(u \otimes 1) = \bar{T}$. Moreover, using $\Omega 0$, we can change $(1 \otimes u)T^{-1}(1 \otimes u)$ to $(u \otimes 1)T^{-1}(u \otimes 1) = \bar{T}$.
We will hold the convention that the left vertical strand corresponds to the first tensor factor, the right vertical strand to the second factor, and the horizontal strand to the third factor.

We now briefly explain the notation. Recall that $T \in \mathcal{H}(H)^{\otimes 2}$ is defined by $T = \sum_i e_i \otimes e^i$. It is a sum of tensor products. Using Sweedler notation we can write $T = T_1 \otimes T_2$. Similarly, we will write $T^{-1} = T_1^{-1} \otimes T_2^{-1}$. The numbers 1 and 2 denote “first tensor element" and “second tensor element." They can also be interpreted as follows: 1 means “under strand” and the 2 means “over strand.” If we have a singular crossing, then we can also denote it with more standard tensor notation. If we write T_{32}, then the 3 and 2 correspond to the third and second tensor element. Their positions imply which one is the under, and which one the over strand. The first number is always the under strand. Thus $T_{32} = 1 \otimes T_2 \otimes T_1 \in \mathcal{H}(H)^{\otimes 3}$. Similarly $T_{32}^{-1} = 1 \otimes T_2^{-1} \otimes T_1^{-1} \in \mathcal{H}(H)^{\otimes 3}$. Lastly, recall that we use shorthand notation for u. We write u to mean $(1 \otimes u) \in \mathcal{H}(H)$.
We’ll will again provide a picture of the moves, and their corresponding equations. The equations are

$$\begin{aligned}
&\text{MP 1.1} : T_{13}T_{23} = T_{21}T_{13}T_{21}^{-1}, \\
&\text{MP 1.2} : T_{23}^{-1}T_{13}^{-1} = T_{21}T_{13}^{-1}T_{21}^{-1}, \\
&\text{MP 1.3} : T_{13}T_{23}^{-1} = T_2^{-1}T_1uT_2u \otimes T_1T_1^{-1} \otimes T_2, \\
&\text{MP 1.4} : T_{23}T_{13}^{-1} = uT_2^{-1}uT_1^{-1}T_2 \otimes T_1T_1^{-1} \otimes T_2^{-1}, \\
&\text{MP 2.1} : T_{31}^{-1}T_{32}^{-1} = T_{12}^{-1}T_{31}^{-1}T_{12}, \\
&\text{MP 2.2} : T_{32}T_{31} = T_{12}^{-1}T_{31}T_{12}, \\
&\text{MP 2.3} : T_{31}^{-1}T_{32} = T_1T_2^{-1}uT_1^{-1}u \otimes T_2^{-1}T_2 \otimes T_1^{-1}, \\
&\text{MP 2.4} : T_{32}^{-1}T_{31} = uT_1uT_2T_1^{-1} \otimes T_2^{-1}T_2 \otimes T_1, \\
&\text{MP 3.1} : T_{31}^{-1}T_{23} = T_{21}T_{23}T_{31}^{-1}, \\
&\text{MP 3.2} : T_{31}^{-1}T_{23}^{-1} = T_{23}^{-1}T_{21}^{-1}T_{31}^{-1}, \\
&\text{MP 3.3} : T_{31}T_{23} = T_2uT_2^{-1}u \otimes T_1^{-1}T_1 \otimes T_2T_1, \\
&\text{MP 3.4} : T_{31}T_{23}^{-1} = T_{23}^{-1}T_{31}T_{21}, \\
&\text{MP 4.1} : T_{23}^{-1}T_{31} = T_{31}T_{23}^{-1}T_{21}^{-1}, \\
&\text{MP 4.2} : T_{23}T_{31} = T_{31}T_{21}T_{23}, \\
&\text{MP 4.3} : T_{23}^{-1}T_{31}^{-1} = uT_2uT_2^{-1} \otimes T_1^{-1}T_1 \otimes T_1^{-1}T_2^{-1}, \\
&\text{MP 4.4} : T_{23}T_{31}^{-1} = T_{21}^{-1}T_{31}^{-1}T_{23}.
\end{aligned}$$



Note: if one were to look at Mihalache, Suzuki, and Terashima's version of the equation of MP 3.4 in [7], then be aware that there is an error. On the left hand side, $\overline{T}_{31}$ should be T_{31}.

Most moves easily transform to MP 4.2, while the ones involving u might need a bit more work and the optional use of $\Omega 0$. For example, the top crossing in the RHS of MP 2.3 cannot be interpreted as $\overline{T}$, but after applying $\Omega 0$ to it, it can be interpreted as such. We have had some difficulties with showing all the transformations, but all MP moves were checked for the Radford algebra with the help of Mathematica code. The scripts for checking MP 4.2, MP 1.3 and MP 2.3 are included in the appendix.

7.3 The Pontrjagin move

The Pontrjagin move is also called the 3-5 move. There are two version of it for spin manifolds. We will provide a proof that our construction is invariant under the Pontrjagin move by a similar technique that Mihalache, Suzuki and Terashima did in [7] for just compact oriented closed manifolds. This is at the same time the most innovating part of our invariant, as the same authors have given an invariant of framed manifolds. Framed manifolds do not require invariance under the 3-5 move.

We will do the proof with tensor diagrams. It is advised to read Section 6.2 to understand the notation of the graphical proof.

We will start off with a small diagram that is often used, namely "the loop." By using a small of abuse of notation to go from the o-graph to the tensor diagram directly, we obtain the Figure 31.
We will prove that the loop is the is the map $x \mapsto \langle \omega_L, x \rangle e_R$.

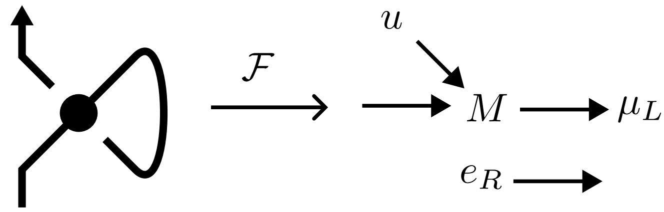
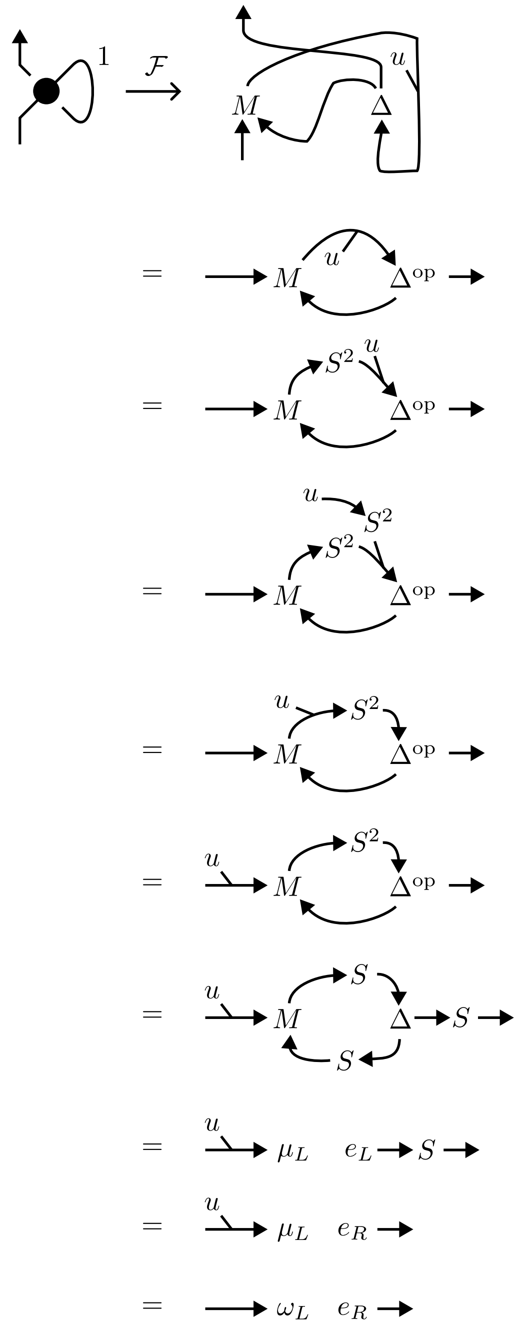


Figure 31: The tensor diagram of the loop.

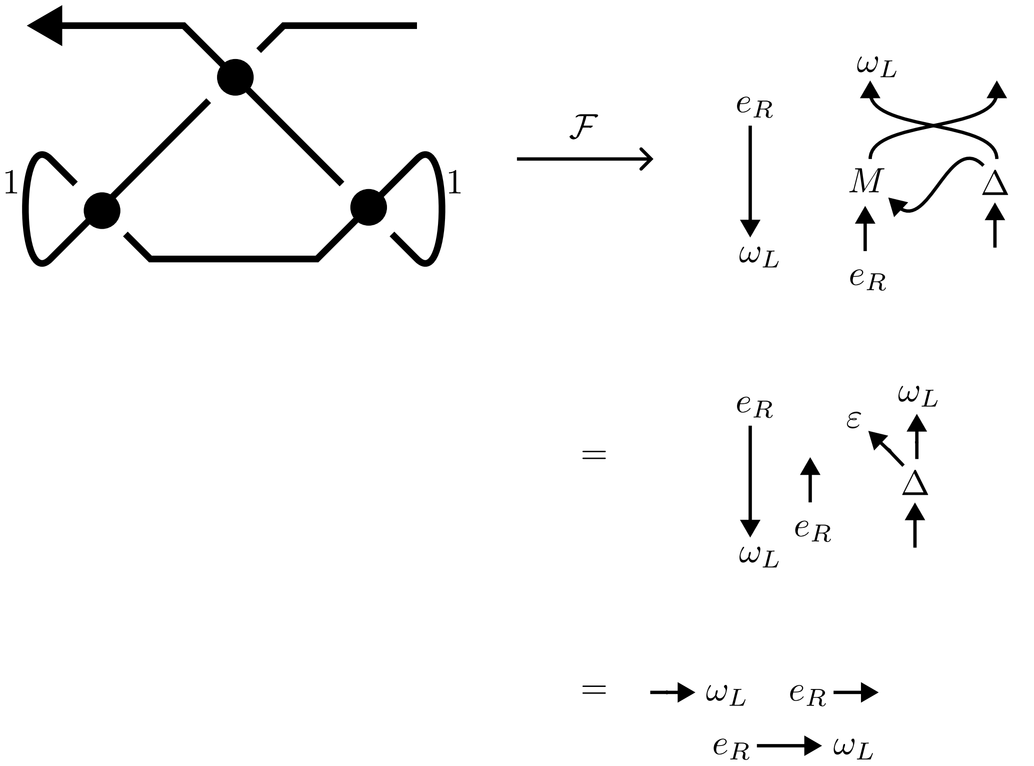
Proof. The prove is done graphically via tensor diagrams. We use $xu = uS^2(x)$ and the fact that $S^2(u) = u$. Recall that we have removed the stars on all the maps, and that these are the multiplication, comultiplication and antipode maps in H^*. Also note that we use μ_L and e_L as a left integral and cointegral such that $\mu_L(e_L) = 1$. We set $e_R := S(e_L)$.



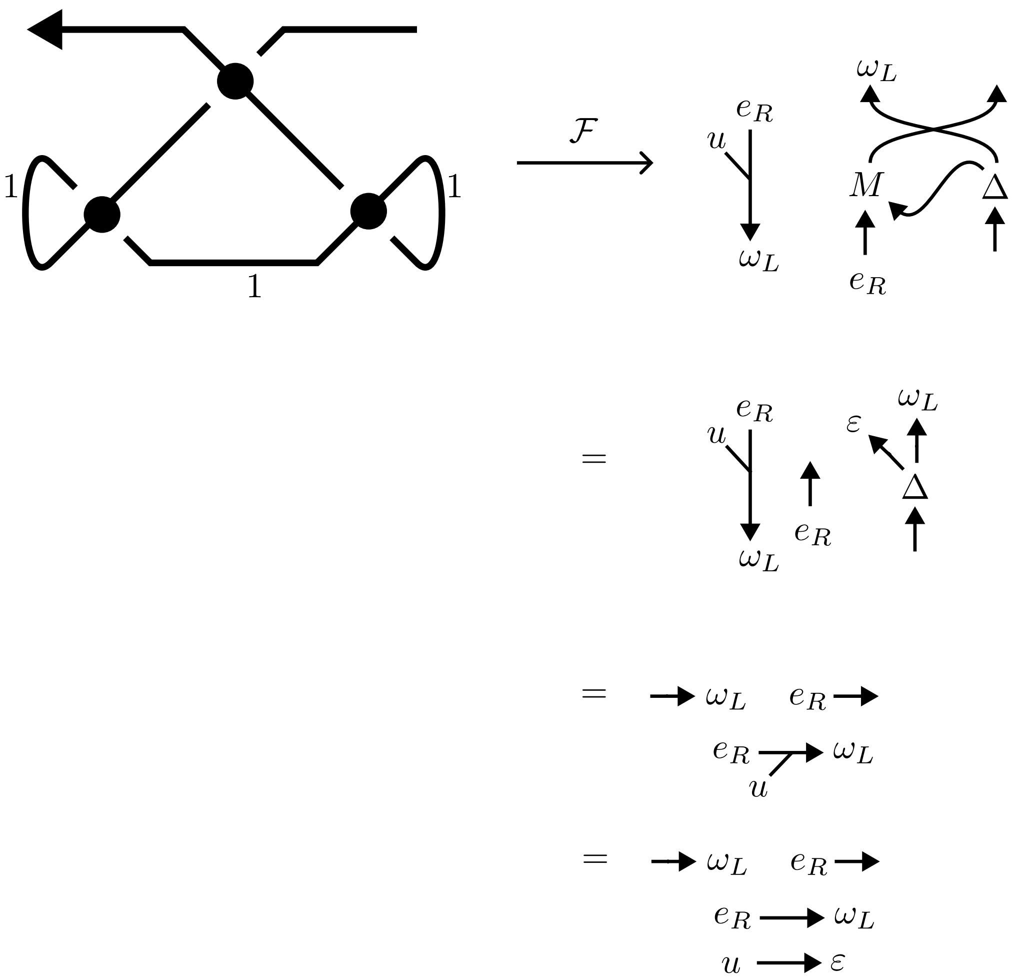
□

Now we want to prove the equivalence of the full Pontrjagin move. This proof is done in a few steps. First, we make the tensor diagrams of the left hand sides of the 3-5 moves. We use the diagram for the loop two times to speed up the process and afterwards we try to simplify the diagrams as much as possible. We then move on the the right hand sides. We cut the right hand sides of the 3-5 moves to exclude the top vertex, compute their tensor diagrams, and simplify them as much as possible. Lastly, we reattach the top vertex and link the previously obtained diagrams to the diagram of a single crossing in the right way. We thus obtain two pairs of diagrams. The LHS and RHS diagrams for the 3-5 move with the extra spin, and the LHS and RHS diagrams for the 3-5 move without the extra spin. We then want to show that the corresponding diagrams are equivalent.

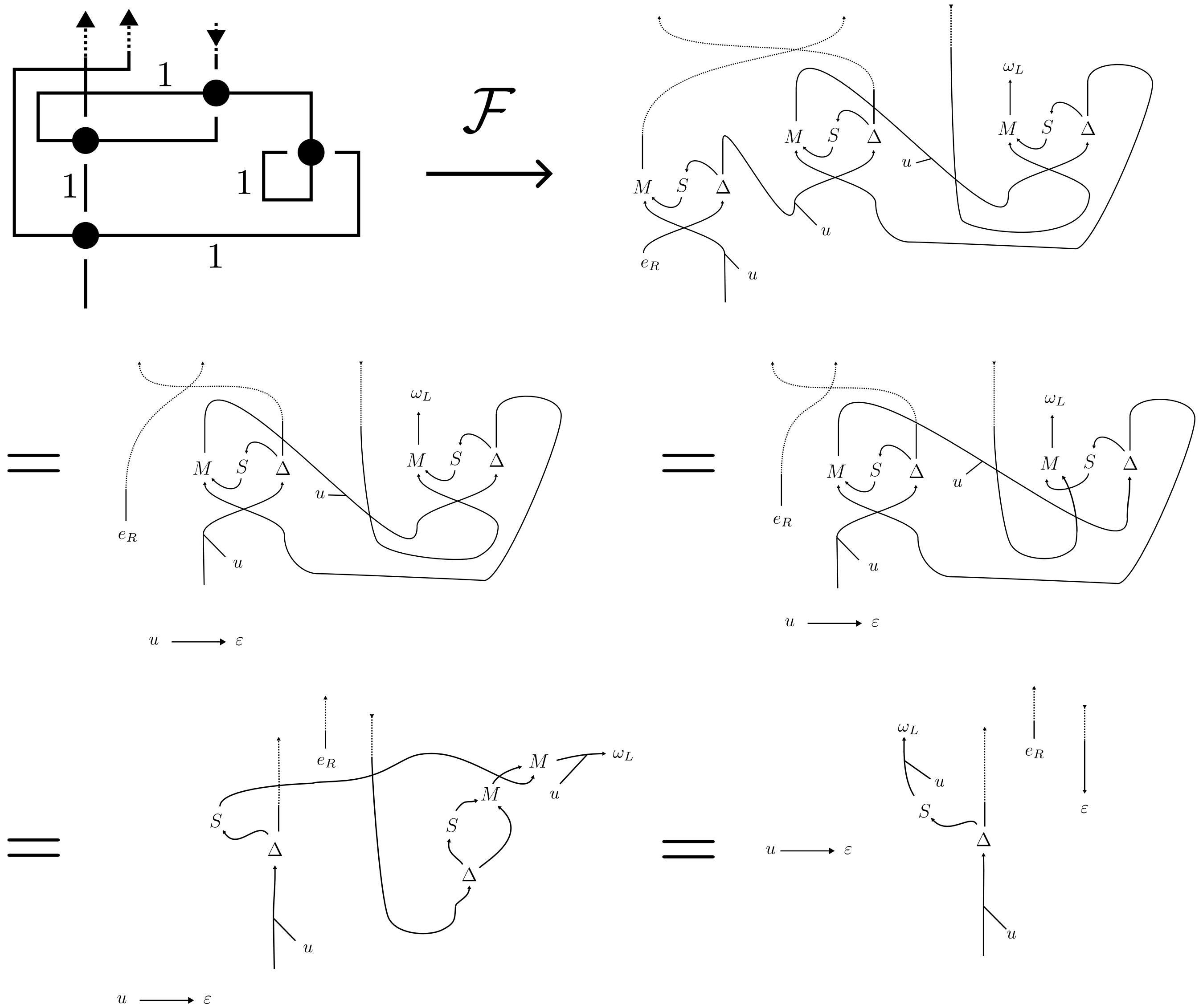
Proof.

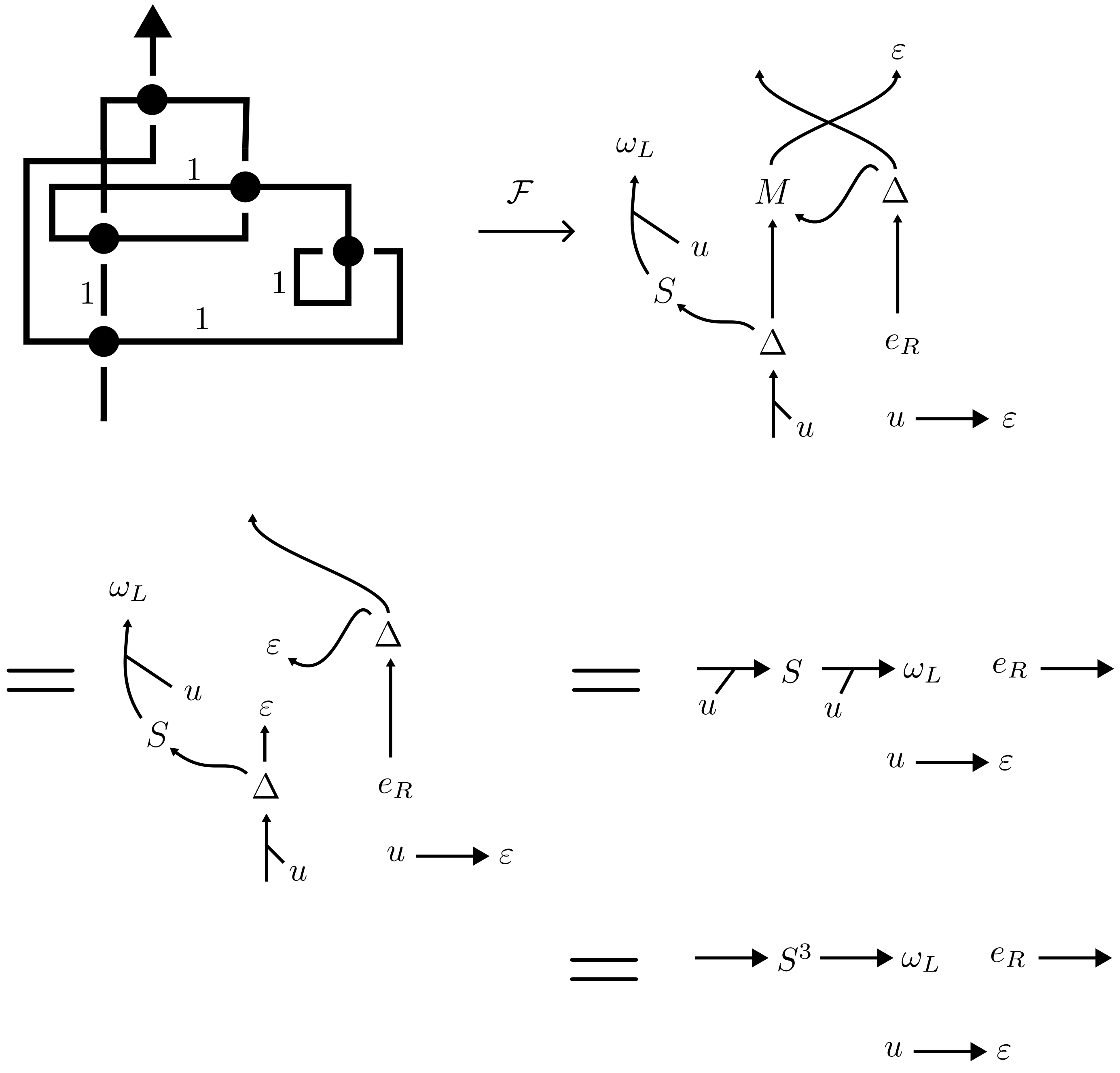


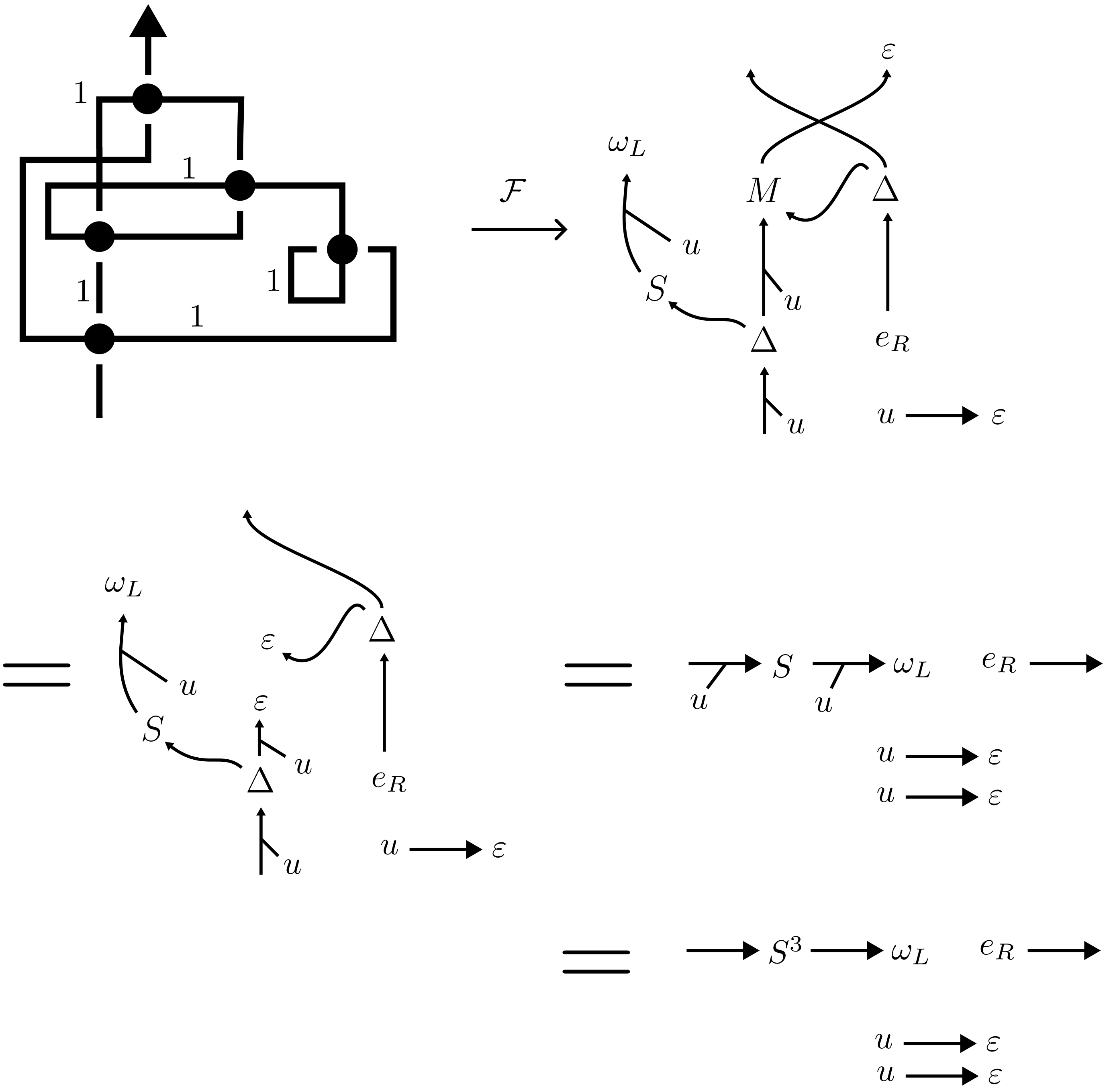
1
1
$\mathcal{F}$
e_R
ω_L
ω_L
M
Δ
e_R
=
e_R
ω_L
ε
Δ
ω_L
e_R
=
ω_L e_R
e_R ω_L



1
1
1
$\mathcal{F}$
e_R
u
ω_L
ω_L
M
Δ
e_R
=
e_R
u
ω_L
ε
Δ
ω_L
e_R
=
ω_L e_R
e_R ω_L
u
=
ω_L e_R
e_R ω_L
u ε


1
1
1
1
𝓕
M
S
Δ
ω_L
e_R
u
u → ε


1
1
1
1
$\mathcal{F}$
ω_L
u
S
M
Δ
ε
e_R
u
$u \to \varepsilon$
$=$
ω_L
u
S
ε
Δ
ε
Δ
e_R
u
$u \to \varepsilon$
$=$
S
u
ω_L
u
e_R
$u \to \varepsilon$
$=$
S^3
ω_L
e_R
$u \to \varepsilon$



□

Perhaps one step that might not be immediately obvious, is how we got for $S(xu)u$ to $S^3(x)$. This step can be seen as follows $S(xu)u = S(u)S(x)u = S(u)uS^3(x) = uuS^3(x) = S^3(x)$. In the below diagrams we also made heavy use of all the lemmas in Section 5.2. A few result to recall are $\omega_L(ab) = \omega_L(ba)$, $\omega_L(S(a)) = \omega_L(a)$, $\varepsilon(u) = 1$, and $\omega_L(e_R) = 1$. This then, for example, yield Figure 32.

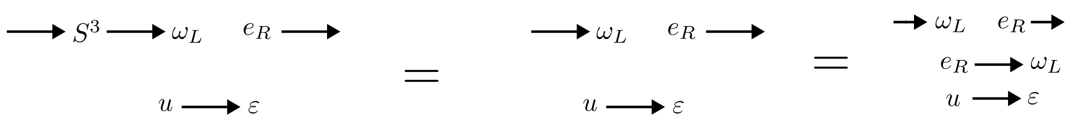


Figure 32

8 Other representations

In our search for invariants, we have tried other representations as well. For instance, we have tried to find an invariant for just compact, connected, oriented 3-manifolds without any extra structure. If the manifolds is not closed, then it is encoded by something we will call a bare o-graph. This is an o-graph as in Definition 2.1.4 in [2]. They require invariance under a move called the C-move, seen in Figure 33. For now do not pay any attention to the colors. The main thing that should be remarked about this move, is that the beginning and end points do not stay consistent. This means that any attempt at an invariant should deal with links. The second problem with this is that there is no notion of direction on a bare o-graph. We tried to circumvent this by finding an invariant that does not depend on the direction in which one multiplies elements.

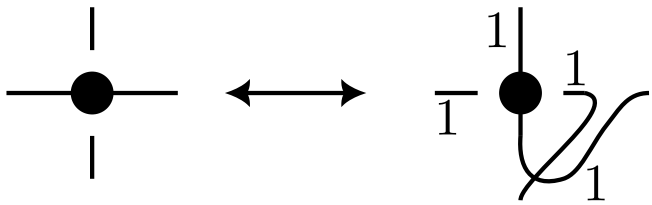


Figure 33: The C-move for bare o-graphs. Note: the colors should not be interpreted as framing or spin numbers. They are rotations of triple lines.

In order for us to work with the C-move, we tried to look at something a bit more stronger than a representation, we want a Hopf-module. This is a module, and a comodule over $\mathcal{H}(H)$.
This is the diagram that a Hopf module should satisfy. This can be found in Sweedler's book [12].

$$\begin{array}{ccccc} M \otimes H & \longrightarrow & M & \xrightarrow{\rho} & M \otimes H \\ \Big\downarrow{\scriptstyle \rho\otimes\Delta} & & & & \Big\uparrow{\scriptstyle \cdot\otimes M} \\ M \otimes H \otimes H \otimes H & & \xrightarrow{\mathrm{Id}\otimes T\otimes \mathrm{Id}} & & M \otimes H \otimes H \otimes H \end{array}$$

Here, ρ the comodule structure. Unfortunately, this diagram only works for representations of Hopf algebras. The Heisenberg double $\mathcal{H}(H)$ is not a Hopf algebra, so we need to check an other requirement.

We tried to find a representation that works for the C-move. We could for example solve for the C-move by looking at representations which are dual to themselves. One way of defining a dual of a representation is as

$$\rho^* : A \to \mathrm{End(V)}$$
$$\rho^*(a) = \rho(S(a)).$$

This approach did not yield any meaningful results.

9 Differences with previous findings and future recommendations

Serban Matei Mihalache, Sakie Suzuki and Yuji Terashima have published their paper titled "Quantum Invariants of Closed Framed 3-Manifolds Based on Ideal Triangulations" [6]. In this paper they come up with an invariant for framed 3-manifolds by using an element they call G, which they define as $G := \sum_{i,j} S^2(e_i)S^{-1}(e_j) \otimes e^j e^i$, which plays the role that u would have done for us. This element satisfies $G(a \otimes f)G^{-1} = s^2(a) \otimes (S^*)^{-2}(f)$ for $a \otimes f \in \mathcal{H}(H)$. The main difference is that they are able to solve the $\Omega 0$ move using this, and do not need to solve the 3-5 move. We have looked to see whether G would work with the 3-5 move, but the main issue why it does not work is the following.

With our element u we had that it was purely an element of H^*. This meant that in the tensor diagram of the Fock representation, the element $1 \otimes u \in \mathcal{H}(H)$ became just multiplying from the right by u. In other words, the tensor diagram for u became just Figure 22. On first glance, we see that G is does not immediately become just multiplication by some element. There still persists some comultiplication after doing the Fock representation. We have checked with some Mathematica code if there is a single element in H^* such that its Fock representation is exactly the same as that of G, but at least for the Radford algebra, there is no such element. This means that our element u really differs from the element G that Mihalache, Suzuki and Terashima found in [6], since it is able to solve the 3-5 move.

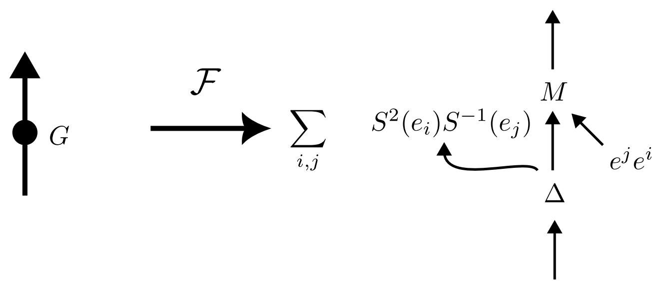


Figure 34: The tensor diagram for G. Compare this with Figure 22.

For future research; one immediate obstacle that we would like to clear is technological. The code given in the appendix takes a long time to run for rather simple calculations. We would like to see this code written in a different language that might be more optimized to deal with this. Specifically, a language optimized for tensor multiplication. We have also not calculated the matrices for T_1, T_2, T_1^{-1}, and T_2^{-1}. We think that calculating them first should speed up the process considerably, as one only needs to do matrix multiplication afterwards to obtain the almost invariant.

References

[1] Nicolas Andruskiewitsch and Walter Ferrer Santos. *The beginnings of the theory of Hopf algebras*. 2009. arXiv: 0901.2460 [math.HO]. URL: https://arxiv.org/abs/0901.2460.

[2] R Benedetti and Carlo Petronio. *Branched standard spines of 3-manifolds*. English. Berlin SE - viii, 132 pages : illustrations ; 24 cm: Springer, 1997. ISBN: 3540626271; 9783540626275. DOI: LK-https://rug.on.worldcat.org/oclc/36430685.

[3] B. G. Casler. "An Imbedding Theorem for Connected 3-Manifolds with Boundary". In: *Proceedings of the American Mathematical Society* 16.4 (1965), pp. 559–566. ISSN: 00029939, 10886826. URL: http://www.jstor.org/stable/2033878 (visited on 09/07/2025).

[4] Greg Kuperberg. *Non-involutory Hopf algebras and 3-manifold invariants*. 1997. arXiv: q-alg/9712047 [q-alg]. URL: https://arxiv.org/abs/q-alg/9712047.

[5] S V Matveev. *Algorithmic topology and classification of 3-manifolds*. English. Berlin SE - xii, 478 pages : illustrations ; 25 cm: Springer, 2003. ISBN: 3540441719; 9783540441717. DOI: LK-https://rug.on.worldcat.org/oclc/51270499.

[6] Serban Matei Mihalache, Sakie Suzuki, and Yuji Terashima. *Quantum invariants of closed framed* 3-*manifolds based on ideal triangulations*. 2022. arXiv: 2209.07378 [math.GT]. URL: https://arxiv.org/abs/2209.07378.

[7] Serban Matei Mihalache, Sakie Suzuki, and Yuji Terashima. "The Heisenberg double of involutory Hopf algebras and invariants of closed 3–manifolds". In: *Algebraic & Geometric Topology* 24.7 (Dec. 2024), pp. 3669–3691. ISSN: 1472-2747. DOI: 10.2140/agt.2024.24.3669. URL: http://dx.doi.org/10.2140/agt.2024.24.3669.

[8] J Milnor. "Spin structures on manifolds". In: *Collected Papers of John Milnor: Differential topology* 3 (2007), p. 293.

[9] David E Radford. "The Order of the Antipode of a Finite Dimensional Hopf Algebra is Finite". In: *American Journal of Mathematics TA - TT -* 98.2 (1976), pp. 333–355. DOI: 10.2307/2373888LK-https://rug.on.worldcat.org/oclc/5545489135. URL: https://dx.doi.org/10.2307/2373888.

[10] David E T A - T T - Radford. *Hopf Algebras*. English. Singapore, 2011. DOI: LK-https://rug.on.worldcat.org/oclc/794328359. URL: https://search.ebscohost.com/login.aspx?direct=true%7B%5C&%7Dscope=site%7B%5C&%7Ddb=nlebk%7B%5C&%7Ddb=nlabk%7B%5C&%7DAN=457156.

[11] Sakie Suzuki. *Quantum invariants based on ideal triangulations*. 2023. URL: https://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/2263-06.pdf.

[12] Moss E Sweedler. *Hopf algebras*. English. New York SE - 336 pages 23 cm: W.A. Benjamin, 1969. ISBN: 0805392548; 9780805392548; 9780805392555; 0805392556. DOI: LK-https://rug.on.worldcat.org/oclc/33677.

10 Appendix: Mathematica code

The code is mainly split up in two sections. An initialization block containing all useful functions and definitions, and a second set of smaller scripts that we ran to check for correctness. I want to thank Roland and his at the time PhD student Gorge for the basis of this code.

10.1 Initialization cell text from

```
 (*Note: never use x or y as variable names.*)
Subscript[Base, i_] := {1, Subscript[g, i], Subscript[x, i],
  Subscript[y, i], Subscript[gx, i], Subscript[gy, i], Subscript[xy,
  i], Subscript[gxy, i]}
SimpO[Z_] :=
 Z /. {a_*Subscript[G, i_] :> a*(1 - Subscript[W, i]),
    Subscript[G, i_] :> (1 - Subscript[W, i])} // Expand

Subscript[um, i_, j_ -> k_][Z_] := Expand[Z] /. {
     Subscript[g, i] Subscript[g, j] -> 1,
     Subscript[g, i] Subscript[x, j] -> Subscript[gx, k],
     Subscript[g, i] Subscript[y, j] -> Subscript[gy, k],
     Subscript[g, i] Subscript[gx, j] -> Subscript[x, k],
     Subscript[g, i] Subscript[gy, j] -> Subscript[y, k],
     Subscript[g, i] Subscript[xy, j] -> Subscript[gxy, k],
     Subscript[g, i] Subscript[gxy, j] -> Subscript[xy, k],
     Subscript[x, i] Subscript[g, j] ->
      Subscript[g, k] - 1 - Subscript[gx, k],
     Subscript[x, i] Subscript[x, j] -> Subscript[x, k],
     Subscript[x, i] Subscript[y, j] -> Subscript[xy, k],
     Subscript[x, i] Subscript[gx, j] -> -Subscript[x, k],
     Subscript[x, i] Subscript[gy, j] ->
      Subscript[gy, k] - Subscript[y, k] - Subscript[gxy, k],
     Subscript[x, i] Subscript[xy, j] -> Subscript[xy, k],
     Subscript[x, i] Subscript[gxy, j] -> -Subscript[xy, k],
     Subscript[y, i] Subscript[g, j] ->
      Subscript[g, k] - 1 - Subscript[gy, k],
     Subscript[y, i] Subscript[x, j] ->
      Subscript[x, k] + Subscript[y, k] - Subscript[xy, k],
     Subscript[y, i] Subscript[y, j] -> Subscript[y, k],
     Subscript[y, i] Subscript[gx, j] -> -Subscript[x, k] - Subscript[
       gy, k] + Subscript[gxy, k],
     Subscript[y, i] Subscript[gy, j] -> -Subscript[y, k],
     Subscript[y, i] Subscript[xy, j] -> Subscript[y, k],
     Subscript[y, i] Subscript[gxy, j] ->
      Subscript[gxy, k] - Subscript[xy, k] - Subscript[gy, k],
     Subscript[gx, i] Subscript[g, j] ->
      1 - Subscript[g, k] - Subscript[x, k],
     Subscript[gx, i] Subscript[x, j] -> Subscript[gx, k],
     Subscript[gx, i] Subscript[y, j] -> Subscript[gxy, k],
     Subscript[gx, i] Subscript[gx, j] -> -Subscript[gx, k],
     Subscript[gx, i] Subscript[gy, j] ->
      Subscript[y, k] - Subscript[gy, k] - Subscript[xy, k],
     Subscript[gx, i] Subscript[xy, j] -> Subscript[gxy, k],
     Subscript[gx, i] Subscript[gxy, j] -> -Subscript[gxy, k],
     Subscript[gy, i] Subscript[g, j] ->
      1 - Subscript[g, k] - Subscript[y, k],
     Subscript[gy, i] Subscript[x, j] ->
      Subscript[gx, k] + Subscript[gy, k] - Subscript[gxy, k],
     Subscript[gy, i] Subscript[y, j] -> Subscript[gy, k],
     Subscript[gy, i] Subscript[gx, j] -> -Subscript[gx, k] -
       Subscript[y, k] + Subscript[xy, k],
     Subscript[gy, i] Subscript[gy, j] -> -Subscript[gy, k],
```

```
    Subscript[gy, i] Subscript[xy, j] -> Subscript[gy, k],
    Subscript[gy, i] Subscript[gxy, j] -> -Subscript[gxy, k] -
      Subscript[y, k] + Subscript[xy, k],
    Subscript[xy, i] Subscript[g, j] ->
     Subscript[gxy, k] + Subscript[y, k] - Subscript[gy, k] -
      Subscript[x, k] - Subscript[gx, k] - 1 + Subscript[g, k],
    Subscript[xy, i] Subscript[x, j] -> Subscript[x, k],
    Subscript[xy, i] Subscript[y, j] -> Subscript[xy, k],
    Subscript[xy, i] Subscript[gx, j] -> -Subscript[x, k] -
      Subscript[gy, k] + Subscript[y, k] - Subscript[xy, k] +
      Subscript[gxy, k],
    Subscript[xy, i] Subscript[gy, j] -> -Subscript[xy, k],
    Subscript[xy, i] Subscript[xy, j] -> Subscript[xy, k],
    Subscript[xy, i] Subscript[gxy, j] -> -2 Subscript[xy, k] -
      Subscript[gy, k] + Subscript[y, k] + Subscript[gxy, k],
    Subscript[gxy, i] Subscript[g, j] ->
     Subscript[xy, k] + Subscript[gy, k] - Subscript[y, k] -
      Subscript[gx, k] - Subscript[x, k] - Subscript[g, k] + 1,
    Subscript[gxy, i] Subscript[x, j] -> Subscript[gx, k],
    Subscript[gxy, i] Subscript[y, j] -> Subscript[gxy, k],
    Subscript[gxy, i] Subscript[gx, j] -> -Subscript[gx, k] -
      Subscript[y, k] + Subscript[gy, k] - Subscript[gxy, k] +
      Subscript[xy, k],
    Subscript[gxy, i] Subscript[gy, j] -> -Subscript[gxy, k],
    Subscript[gxy, i] Subscript[xy, j] -> Subscript[gxy, k],
    Subscript[gxy, i] Subscript[gxy, j] -> -2 Subscript[gxy, k] -
      Subscript[y, k] + Subscript[gy, k] + Subscript[xy, k]
    } /. {Subscript[x_, i] :> Subscript[x, k],
   Subscript[x_, j] :> Subscript[x, k]} // Expand

Subscript[u\[CapitalDelta], i_ -> j_, k_][Z_] := Expand[Z] /. {
   Subscript[g, i] -> Subscript[g, j] Subscript[g, k],
   Subscript[x, i] ->
    Subscript[x, j] Subscript[g, k] + Subscript[x, k],
   Subscript[y, i] ->
    Subscript[y, j] Subscript[g, k] + Subscript[y, k],
   Subscript[gx, i] ->
    Subscript[gx, j] + Subscript[g, j] Subscript[gx, k],
   Subscript[gy, i] ->
    Subscript[gy, j] + Subscript[g, j] Subscript[gy, k],
   Subscript[xy, i] ->
    Subscript[xy, j] + Subscript[x, j] Subscript[gy, k] - Subscript[
     y, j] + Subscript[y, j] Subscript[g, k] -
     Subscript[y, j] Subscript[gx, k] + Subscript[xy, k],
   Subscript[gxy, i] ->
    Subscript[gxy, j] Subscript[g, k] +
     Subscript[gx, j] Subscript[y, k] -
     Subscript[gy, j] Subscript[g, k] + Subscript[gy, j] -
     Subscript[gy, j] Subscript[x, k] +
     Subscript[g, j] Subscript[gxy, k]
   } // Expand

Subscript[u\[Epsilon], i_][Z_] := Expand[Z] /. {
  Subscript[g, i] -> 1,
  Subscript[x, i] -> 0,
  Subscript[y, i] -> 0,
  Subscript[gx, i] -> 0,
  Subscript[gy, i] -> 0,
  Subscript[xy, i] -> 0,
  Subscript[gxy, i] -> 0
  }
```

```
Subscript[uS, i_][Z_] := Expand[Z] /. {
     Subscript[g, i] -> Subscript[g, i],
     Subscript[x, i] -> 1 - Subscript[g, i] + Subscript[gx, i],
     Subscript[y, i] -> 1 - Subscript[g, i] + Subscript[gy, i],
     Subscript[gx, i] -> -Subscript[x, i],
     Subscript[gy, i] -> -Subscript[y, i],
     Subscript[xy, i] ->
      1 - Subscript[g, i] + Subscript[gy, i] - Subscript[x, i] +
       Subscript[xy, i],
     Subscript[gxy, i] ->
      Subscript[gxy, i] - Subscript[gy, i] - Subscript[x, i]
     } // Expand

Subscript[
\!\(\*OverscriptBox[\(uS\), \(_\)]\), i_][Z_] := Expand[Z] /. {
     Subscript[g, i] -> Subscript[g, i],
     Subscript[x, i] -> -Subscript[gx, i],
     Subscript[y, i] -> -Subscript[gy, i],
     Subscript[gx, i] -> -1 + Subscript[g, i] + Subscript[x, i],
     Subscript[gy, i] -> -1 + Subscript[g, i] + Subscript[y, i],
     Subscript[xy,
      i] -> -Subscript[gx, i] + Subscript[xy, i] - Subscript[y, i],
     Subscript[gxy,
      i] -> -1 + Subscript[g, i] - Subscript[gx, i] + Subscript[gxy,
        i] + Subscript[y, i]
     } // Expand
(*Dual*)
Subscript[OBase, i_] := {Subscript[W, i], Subscript[G, i], Subscript[
  X, i], Subscript[Y, i], Subscript[GX, i], Subscript[GY, i],
  Subscript[XY, i], Subscript[GXY, i]}

Subscript[Pairing, i_, j_][Z_] := Expand[Z] /. {
    Subscript[g, i] Subscript[W, j] -> 0,
    Subscript[g, i] Subscript[G, j] -> 1,
    Subscript[g, i] Subscript[X, j] -> 0,
    Subscript[g, i] Subscript[Y, j] -> 0,
    Subscript[g, i] Subscript[GX, j] -> 0,
    Subscript[g, i] Subscript[GY, j] -> 0,
    Subscript[g, i] Subscript[XY, j] -> 0,
    Subscript[g, i] Subscript[GXY, j] -> 0,
    Subscript[x, i] Subscript[W, j] -> 0,
    Subscript[x, i] Subscript[G, j] -> 0,
    Subscript[x, i] Subscript[X, j] -> 1,
    Subscript[x, i] Subscript[Y, j] -> 0,
    Subscript[x, i] Subscript[GX, j] -> 0,
    Subscript[x, i] Subscript[GY, j] -> 0,
    Subscript[x, i] Subscript[XY, j] -> 0,
    Subscript[x, i] Subscript[GXY, j] -> 0,
    Subscript[y, i] Subscript[W, j] -> 0,
    Subscript[y, i] Subscript[G, j] -> 0,
    Subscript[y, i] Subscript[X, j] -> 0,
    Subscript[y, i] Subscript[Y, j] -> 1,
    Subscript[y, i] Subscript[GX, j] -> 0,
    Subscript[y, i] Subscript[GY, j] -> 0,
    Subscript[y, i] Subscript[XY, j] -> 0,
    Subscript[y, i] Subscript[GXY, j] -> 0,
    Subscript[gx, i] Subscript[W, j] -> 0,
    Subscript[gx, i] Subscript[G, j] -> 0,
    Subscript[gx, i] Subscript[X, j] -> 0,
    Subscript[gx, i] Subscript[Y, j] -> 0,
```

```
    Subscript[gx, i] Subscript[GX, j] -> 1,
    Subscript[gx, i] Subscript[GY, j] -> 0,
    Subscript[gx, i] Subscript[XY, j] -> 0,
    Subscript[gx, i] Subscript[GXY, j] -> 0,
    Subscript[gy, i] Subscript[W, j] -> 0,
    Subscript[gy, i] Subscript[G, j] -> 0,
    Subscript[gy, i] Subscript[X, j] -> 0,
    Subscript[gy, i] Subscript[Y, j] -> 0,
    Subscript[gy, i] Subscript[GX, j] -> 0,
    Subscript[gy, i] Subscript[GY, j] -> 1,
    Subscript[gy, i] Subscript[XY, j] -> 0,
    Subscript[gy, i] Subscript[GXY, j] -> 0,
    Subscript[xy, i] Subscript[W, j] -> 0,
    Subscript[xy, i] Subscript[G, j] -> 0,
    Subscript[xy, i] Subscript[X, j] -> 0,
    Subscript[xy, i] Subscript[Y, j] -> 0,
    Subscript[xy, i] Subscript[GX, j] -> 0,
    Subscript[xy, i] Subscript[GY, j] -> 0,
    Subscript[xy, i] Subscript[XY, j] -> 1,
    Subscript[xy, i] Subscript[GXY, j] -> 0,
    Subscript[gxy, i] Subscript[W, j] -> 0,
    Subscript[gxy, i] Subscript[G, j] -> 0,
    Subscript[gxy, i] Subscript[X, j] -> 0,
    Subscript[gxy, i] Subscript[Y, j] -> 0,
    Subscript[gxy, i] Subscript[GX, j] -> 0,
    Subscript[gxy, i] Subscript[GY, j] -> 0,
    Subscript[gxy, i] Subscript[XY, j] -> 0,
    Subscript[gxy, i] Subscript[GXY, j] -> 1, Subscript[W, j] -> 1,
    Subscript[G, j] -> 0, Subscript[X, j] -> 0, Subscript[Y, j] -> 0,
    Subscript[GX, j] -> 0, Subscript[GY, j] -> 0,
    Subscript[XY, j] -> 0, Subscript[GXY, j] -> 0,
    Subscript[g, i] -> 1, Subscript[x, i] -> 0, Subscript[y, i] -> 0,
    Subscript[gx, i] -> 0, Subscript[gy, i] -> 0,
    Subscript[xy, i] -> 0, Subscript[gxy, i] -> 0}
Subscript[om, i_, j_ -> k_][Z_] := Expand[Z] /. {
      Subscript[W, i] Subscript[W, j] -> Subscript[W, k],
      Subscript[G, i] Subscript[W, j] -> 0,
      Subscript[X, i] Subscript[W, j] -> 0,
      Subscript[Y, i] Subscript[W, j] -> -Subscript[XY, k],
      Subscript[GX, i] Subscript[W, j] -> Subscript[GX, k],
      Subscript[GY, i] Subscript[W, j] ->
        Subscript[GXY, k] + Subscript[GY, k],
      Subscript[XY, i] Subscript[W, j] -> Subscript[XY, k],
      Subscript[GXY, i] Subscript[W, j] -> 0,
      Subscript[W, i] Subscript[G, j] -> 0,
      Subscript[G, i] Subscript[G, j] -> Subscript[G, k],
      Subscript[X, i] Subscript[G, j] -> Subscript[X, k],
      Subscript[Y, i] Subscript[G, j] ->
        Subscript[XY, k] + Subscript[Y, k],
      Subscript[GX, i] Subscript[G, j] -> 0,
      Subscript[GY, i] Subscript[G, j] -> -Subscript[GXY, k],
      Subscript[XY, i] Subscript[G, j] -> 0,
      Subscript[GXY, i] Subscript[G, j] -> Subscript[GXY, k],
      Subscript[W, i] Subscript[X, j] -> Subscript[X, k],
      Subscript[G, i] Subscript[X, j] -> 0,
      Subscript[X, i] Subscript[X, j] -> 0,
      Subscript[Y, i] Subscript[X, j] -> 0,
      Subscript[GX, i] Subscript[X, j] -> 0,
      Subscript[GY, i] Subscript[X, j] -> -Subscript[GXY, k],
      Subscript[XY, i] Subscript[X, j] -> 0,
      Subscript[GXY, i] Subscript[X, j] -> 0,
```

```
    Subscript[W, i] Subscript[Y, j] -> Subscript[Y, k],
    Subscript[G, i] Subscript[Y, j] -> 0,
    Subscript[X, i] Subscript[Y, j] -> 0,
    Subscript[Y, i] Subscript[Y, j] -> 0,
    Subscript[GX, i] Subscript[Y, j] -> Subscript[GXY, k],
    Subscript[GY, i] Subscript[Y, j] -> 0,
    Subscript[XY, i] Subscript[Y, j] -> 0,
    Subscript[GXY, i] Subscript[Y, j] -> 0,
    Subscript[W, i] Subscript[GX, j] -> 0,
    Subscript[G, i] Subscript[GX, j] -> Subscript[GX, k],
    Subscript[X, i] Subscript[GX, j] -> 0,
    Subscript[Y, i] Subscript[GX, j] -> -Subscript[XY, k],
    Subscript[GX, i] Subscript[GX, j] -> 0,
    Subscript[GY, i] Subscript[GX, j] -> 0,
    Subscript[XY, i] Subscript[GX, j] -> 0,
    Subscript[GXY, i] Subscript[GX, j] -> 0,
    Subscript[W, i] Subscript[GY, j] -> 0,
    Subscript[G, i] Subscript[GY, j] -> Subscript[GY, k],
    Subscript[X, i] Subscript[GY, j] -> Subscript[XY, k],
    Subscript[Y, i] Subscript[GY, j] -> 0,
    Subscript[GX, i] Subscript[GY, j] -> 0,
    Subscript[GY, i] Subscript[GY, j] -> 0,
    Subscript[XY, i] Subscript[GY, j] -> 0,
    Subscript[GXY, i] Subscript[GY, j] -> 0,
    Subscript[W, i] Subscript[XY, j] -> Subscript[XY, k],
    Subscript[G, i] Subscript[XY, j] -> 0,
    Subscript[X, i] Subscript[XY, j] -> 0,
    Subscript[Y, i] Subscript[XY, j] -> 0,
    Subscript[GX, i] Subscript[XY, j] -> 0,
    Subscript[GY, i] Subscript[XY, j] -> 0,
    Subscript[XY, i] Subscript[XY, j] -> 0,
    Subscript[GXY, i] Subscript[XY, j] -> 0,
    Subscript[W, i] Subscript[GXY, j] -> 0,
    Subscript[G, i] Subscript[GXY, j] -> Subscript[GXY, k],
    Subscript[X, i] Subscript[GXY, j] -> 0,
    Subscript[Y, i] Subscript[GXY, j] -> 0,
    Subscript[GX, i] Subscript[GXY, j] -> 0,
    Subscript[GY, i] Subscript[GXY, j] -> 0,
    Subscript[XY, i] Subscript[GXY, j] -> 0,
    Subscript[GXY, i] Subscript[GXY, j] -> 0} /. {Subscript[a_, i] :>
     Subscript[a, k], Subscript[a_, j] :> Subscript[a, k]} // Expand

Subscript[o\[CapitalDelta], i_ -> k_, j_][Z_] := Expand[Z] /. {
   Subscript[W, i] ->
    Subscript[W, j] Subscript[W, k] +
     Subscript[G, j] Subscript[G, k] +
     Subscript[G, j] Subscript[GX, k] +
     Subscript[G, j] Subscript[GXY, k] +
     Subscript[G, j] Subscript[GY, k] -
     Subscript[G, j] Subscript[X, k] -
     Subscript[G, j] Subscript[XY, k] -
     Subscript[G, j] Subscript[Y, k],
   Subscript[G, i] ->
    Subscript[G, j] Subscript[W, k] +
     Subscript[W, j] Subscript[G, k] -
     Subscript[G, j] Subscript[GX, k] -
     Subscript[G, j] Subscript[GXY, k] -
     Subscript[G, j] Subscript[GY, k] +
     Subscript[G, j] Subscript[X, k] +
     Subscript[G, j] Subscript[XY, k] +
     Subscript[G, j] Subscript[Y, k],
```

```
Subscript[X, i] ->
 Subscript[G, k] Subscript[GX, j] -
  Subscript[G, j] Subscript[GX, k] -
  Subscript[G, j] Subscript[GXY, k] +
  Subscript[X, j] Subscript[W, k] +
  Subscript[W, j] Subscript[X, k] -
  Subscript[GX, j] Subscript[X, k] +
  Subscript[X, j] Subscript[X, k] -
  Subscript[G, j] Subscript[XY, k] -
  Subscript[GX, j] Subscript[XY, k] +
  Subscript[X, j] Subscript[XY, k] -
  Subscript[GX, j] Subscript[Y, k] +
  Subscript[X, j] Subscript[Y, k],
Subscript[Y,
 i] -> -Subscript[G, j] Subscript[GXY, k] -
  Subscript[GX, j] Subscript[GXY, k] -
  Subscript[GXY, j] Subscript[GXY, k] +
  Subscript[G, k] Subscript[GY, j] +
  Subscript[GX, k] Subscript[GY, j] -
  Subscript[G, j] Subscript[GY, k] -
  Subscript[GX, j] Subscript[GY, k] -
  Subscript[GXY, j] Subscript[GY, k] -
  Subscript[GY, j] Subscript[X, k] +
  Subscript[G, j] Subscript[XY, k] +
  Subscript[GX, j] Subscript[XY, k] +
  Subscript[GXY, j] Subscript[XY, k] +
  Subscript[Y, j] Subscript[W, k] +
  Subscript[W, j] Subscript[Y, k] -
  Subscript[GY, j] Subscript[Y, k] +
  Subscript[X, j] Subscript[Y, k] +
  Subscript[XY, j] Subscript[Y, k] +
  Subscript[Y, j] Subscript[Y, k],
Subscript[GX, i] ->
 Subscript[GX, j] Subscript[W, k] +
  Subscript[W, j] Subscript[GX, k] -
  Subscript[GX, j] Subscript[GX, k] -
  Subscript[G, j] Subscript[GXY, k] -
  Subscript[GX, j] Subscript[GXY, k] -
  Subscript[GX, j] Subscript[GY, k] +
  Subscript[G, k] Subscript[X, j] +
  Subscript[GX, k] Subscript[X, j] +
  Subscript[GXY, k] Subscript[X, j] +
  Subscript[GY, k] Subscript[X, j] -
  Subscript[G, j] Subscript[X, k] -
  Subscript[G, j] Subscript[XY, k],
Subscript[GY, i] ->
 Subscript[G, j] Subscript[GXY, k] +
  Subscript[GX, j] Subscript[GXY, k] +
  Subscript[GXY, j] Subscript[GXY, k] +
  Subscript[GY, j] Subscript[W, k] -
  Subscript[GX, k] Subscript[GY, j] +
  Subscript[W, j] Subscript[GY, k] -
  Subscript[GY, j] Subscript[GY, k] +
  Subscript[GY, k] Subscript[X, j] +
  Subscript[GY, j] Subscript[X, k] +
  Subscript[GY, k] Subscript[XY, j] -
  Subscript[G, j] Subscript[XY, k] -
  Subscript[GX, j] Subscript[XY, k] -
  Subscript[GXY, j] Subscript[XY, k] +
  Subscript[G, k] Subscript[Y, j] +
  Subscript[GY, k] Subscript[Y, j] -
```

```
      Subscript[G, j] Subscript[Y, k] -
      Subscript[GX, j] Subscript[Y, k] -
      Subscript[GXY, j] Subscript[Y, k],
    Subscript[XY, i] ->
     Subscript[G, k] Subscript[GXY, j] +
      Subscript[G, j] Subscript[GXY, k] +
      Subscript[GX, j] Subscript[GXY, k] +
      Subscript[GXY, j] Subscript[GXY, k] -
      Subscript[GX, k] Subscript[GY, j] +
      Subscript[GX, j] Subscript[GY, k] +
      Subscript[GXY, j] Subscript[GY, k] -
      Subscript[GXY, j] Subscript[X, k] +
      Subscript[XY, j] Subscript[W, k] +
      Subscript[X, k] Subscript[XY, j] +
      Subscript[W, j] Subscript[XY, k] -
      Subscript[GX, j] Subscript[XY, k] -
      2*Subscript[GXY, j] Subscript[XY, k] -
      Subscript[GY, j] Subscript[XY, k] +
      Subscript[XY, j] Subscript[XY, k] +
      Subscript[X, k] Subscript[Y, j] +
      Subscript[XY, k] Subscript[Y, j] -
      Subscript[GXY, j] Subscript[Y, k] -
      Subscript[X, j] Subscript[Y, k],
    Subscript[GXY, i] ->
     Subscript[GXY, j] Subscript[W, k] -
      Subscript[GX, k] Subscript[GXY, j] +
      Subscript[W, j] Subscript[GXY, k] -
      Subscript[GX, j] Subscript[GXY, k] -
      2*Subscript[GXY, j] Subscript[GXY, k] -
      Subscript[GXY, k] Subscript[GY, j] -
      Subscript[GXY, j] Subscript[GY, k] -
      Subscript[GY, k] Subscript[X, j] -
      Subscript[GY, j] Subscript[X, k] +
      Subscript[G, k] Subscript[XY, j] +
      Subscript[GX, k] Subscript[XY, j] +
      Subscript[GXY, k] Subscript[XY, j] +
      Subscript[G, j] Subscript[XY, k] +
      Subscript[GX, j] Subscript[XY, k] +
      Subscript[GXY, j] Subscript[XY, k] +
      Subscript[GX, k] Subscript[Y, j] +
      Subscript[GXY, k] Subscript[Y, j] +
      Subscript[GX, j] Subscript[Y, k] +
      Subscript[GXY, j] Subscript[Y, k]
    }
Subscript[o\[Epsilon], i_][Z_] := Expand[Z] /. {
    Subscript[W, i] -> 1,
    Subscript[G, i] -> 0,
    Subscript[X, i] -> 0,
    Subscript[Y, i] -> 0,
    Subscript[GX, i] -> 0,
    Subscript[GY, i] -> 0,
    Subscript[XY, i] -> 0,
    Subscript[GXY, i] -> 0
    }
Subscript[oS, i_][Z_] := Expand[Z] /. {
    Subscript[W, i] ->
     Subscript[W, i] + Subscript[X, i] + Subscript[Y, i] + Subscript[
      XY, i],
    Subscript[G, i] ->
     Subscript[G, i] - Subscript[X, i] - Subscript[Y, i] - Subscript[
      XY, i],
```

```
    Subscript[X,
     i] -> -Subscript[GX, i] - Subscript[XY, i] - Subscript[GXY, i],
    Subscript[Y, i] -> -Subscript[GY, i],
    Subscript[GX, i] -> Subscript[X, i],
    Subscript[GY, i] ->
     Subscript[Y, i] + Subscript[XY, i] - Subscript[GXY, i],
    Subscript[XY, i] -> Subscript[XY, i],
    Subscript[GXY, i] -> Subscript[GXY, i]
    }

Subscript[
\!\(\*OverscriptBox[\(oS\), \(_\)]\), i_][Z_] := Expand[Z] /. {
    Subscript[W,
     i] -> -Subscript[GX, i] - Subscript[GXY, i] - Subscript[GY, i] +
      Subscript[W, i],
    Subscript[G, i] ->
     Subscript[G, i] + Subscript[GX, i] + Subscript[GXY, i] +
      Subscript[GY, i],
    Subscript[X, i] -> Subscript[GX, i],
    Subscript[Y, i] ->
     Subscript[GXY, i] + Subscript[GY, i] - Subscript[XY, i],
    Subscript[GX,
     i] -> -Subscript[GXY, i] - Subscript[X, i] - Subscript[XY, i],
    Subscript[GY, i] -> -Subscript[Y, i],
    Subscript[XY, i] -> Subscript[XY, i],
    Subscript[GXY, i] -> Subscript[GXY, i]}

Subscript[OComp, i_, j_][Z_] :=
 Expand[Z] /. {Subscript[a_, i] Subscript[a_, j] :> 1,
    Subscript[b_, i] Subscript[b_, j] :> 1,
    Subscript[a_, i] Subscript[b_, j] :> 0}

Subscript[FockMatrix, i_][c_] :=
 Module[{o1, u1, cnew, j, k, m, n, o},
  cnew = Subscript[SplitOU, i, o1, u1][c];
  Table[Subscript[OComp, n, o][b*z], {b,
    Table[Subscript[Hm, k, o1 -> n][
      Subscript[Pairing, u1, m][
       Subscript[o\[CapitalDelta], j -> m, k][cnew*a]]], {a ,
      Subscript[OBase, j]}]}, {z, Subscript[OBase, o]}]
  ]

(*Heisenberg double*)

Subscript[SplitOU, i_, j_, k_][Z_] := Expand[Z] /. {
    Subscript[g, i] -> Subscript[g, k],
    Subscript[x, i] -> Subscript[x, k],
    Subscript[y, i] -> Subscript[y, k],
    Subscript[gx, i] -> Subscript[gx, k],
    Subscript[gy, i] -> Subscript[gy, k],
    Subscript[xy, i] -> Subscript[xy, k],
    Subscript[gxy, i] -> Subscript[gxy, k],
    Subscript[W, i] -> Subscript[W, j],
    Subscript[G, i] -> Subscript[G, j],
    Subscript[X, i] -> Subscript[X, j],
    Subscript[Y, i] -> Subscript[Y, j],
    Subscript[GX, i] -> Subscript[GX, j],
    Subscript[GY, i] -> Subscript[GY, j],
    Subscript[XY, i] -> Subscript[XY, j],
    Subscript[GXY, i] -> Subscript[GXY, j]
    }
```

```
Subscript[Hm, i_, j_ -> k_][Z_] :=
 Module[{i1, i2, j1, j2, i11, i12, j21, j22, Znew},
  Znew =
   Subscript[SplitOU, j, j1, j2][
    Subscript[SplitOU, i, i1, i2][Expand[Z]]];
  Expand[Znew] // Subscript[o\[CapitalDelta], i1 -> i11, i12] //
      Subscript[u\[CapitalDelta], j2 -> j21, j22] // Subscript[
     Pairing, j22, i11] // Subscript[um, i2, j21 -> k] // Subscript[
    om, i12, j1 -> k] // Expand
  ](*I would use this, then you don't to add Subscript[W, \
j]+Subscript[G, j] in every expression solely involing Subscript[g, \
i]'s*)

Subscript[DoubleBase, i_] :=
 Table[a*b, {a, Subscript[Base, i]}, {b, Subscript[OBase, i]}] //
  Flatten
Subscript[T, i_, j_] := Total[Subscript[OBase, i]*Subscript[Base, j]]
Subscript[
\!\(\*OverscriptBox[\(T\), \(_\)]\), i_, j_] :=
 Total[Subscript[OBase, i]*Subscript[
\!\(\*OverscriptBox[\(uS\), \(_\)]\), j][ Subscript[Base, j]]]
Subscript[Tinv, i_, j_] :=
 Total[Subscript[OBase, i]*Subscript[uS, j][ Subscript[Base, j]]]

Subscript[U, i_] := -Subscript[G, i] - Subscript[GX, i] - Subscript[
  GXY, i] - Subscript[GY, i] + Subscript[W, i] + Subscript[X, i] +
  Subscript[XY, i] + Subscript[Y, i]
Subscript[u, i_] := Subscript[g, i]
Subscript[uinv, i_] := Subscript[g, i]
Subscript[Uinv, i_] := -Subscript[G, i] - Subscript[GX, i] -
  Subscript[GXY, i] - Subscript[GY, i] + Subscript[W, i] + Subscript[
  X, i] + Subscript[XY, i] + Subscript[Y, i]
Subscript[uInt, i_] :=
 Subscript[xy, i] + Subscript[gxy, i] - Subscript[x, i] - Subscript[
  gx, i]
Subscript[oInt, i_] := Subscript[XY, i]
```

10.2 Other lines to run text form

```
    (*Pairing axiom*)
Table[Subscript[Pairing, 1, 2][a*b], {a, Subscript[Base, 1]}, {b,
   Subscript[OBase, 2]}] // MatrixForm

    (*Multiplication table*)
Table[Subscript[um, 1, 2 -> 3][a*b], {a, Subscript[Base, 1]}, {b,
   Subscript[Base, 2]}] // MatrixForm

    (*Antipode axioms*)
Table[Subscript[um, 2, 3 -> 1][
   Subscript[uS, 2][Subscript[u\[CapitalDelta], 1 -> 2, 3][a]]] -
  Subscript[u\[Epsilon], 1][a], {a, Subscript[Base, 1]}]
Table[Subscript[um, 2, 3 -> 1][
   Subscript[uS, 3][Subscript[u\[CapitalDelta], 1 -> 2, 3][a]]] -
  Subscript[u\[Epsilon], 1][a], {a, Subscript[Base, 1]}]

    (*u\[CapitalDelta] is an algebra morphism*)
Table[Subscript[u\[CapitalDelta], 3 -> 4, 5][
   Subscript[um, 1, 2 -> 3][a*b]] -
  Subscript[um, 5, 7 -> 5][
```

```
  Subscript[um, 4, 6 -> 4][
   Subscript[u\[CapitalDelta], 2 -> 6, 7][
    Subscript[u\[CapitalDelta], 1 -> 4, 5][a*b]]]], {a, Subscript[
  Base, 1]}, {b, Subscript[Base, 2]}]

   (* co-unit + product (the other one is automatically satisfied)*)
Table[Subscript[u\[Epsilon], 3][Subscript[um, 1, 2 -> 3][a*b]] -
  Subscript[u\[Epsilon], 1][a]*Subscript[u\[Epsilon], 2][b], {a,
  Subscript[Base, 1]}, {b, Subscript[Base, 2]}]
(Subscript[u\[CapitalDelta], 1 -> 2, 3][1] - 1*1) // Expand

   (*co-associativity of u\[CapitalDelta]*)
Table[Subscript[u\[CapitalDelta], 2 -> 1, 2][
   Subscript[u\[CapitalDelta], 1 -> 2, 3][a]] -
  Subscript[u\[CapitalDelta], 2 -> 2, 3][
   Subscript[u\[CapitalDelta], 1 -> 1, 2][a]], {a, Subscript[Base,
  1]}]

   (*Associativity of um*)
Table[Subscript[um, 3, 4 -> 5][Subscript[um, 1, 2 -> 3][a*b*c]] -
  Subscript[um, 1, 3 -> 5][Subscript[um, 2, 4 -> 3][a*b*c]], {a,
  Subscript[Base, 1]}, {b, Subscript[Base, 2]}, {c, Subscript[Base,
  4]}]

   (*Antipode has order 4, and not 2.*)
Table[Subscript[uS, 1][
   Subscript[uS, 1][Subscript[uS, 1][Subscript[uS, 1][a]]]] - a, {a,
  Subscript[Base, 1]}]
Table[Subscript[uS, 1][Subscript[uS, 1][a]] - a, {a, Subscript[Base,
  1]}]

   (*Unit in om*)
Table[Subscript[om, 2,
   1 -> 2][(Subscript[W, 1] + Subscript[G, 1])*a] - a, {a, Subscript[
  OBase, 2]}]

   (*Associativity of om*)
Table[Subscript[om, 3, 4 -> 5][Subscript[om, 1, 2 -> 3][a*b*c]] -
  Subscript[om, 1, 3 -> 5][Subscript[om, 2, 4 -> 3][a*b*c]], {a,
  Subscript[OBase, 1]}, {b, Subscript[OBase, 2]}, {c, Subscript[OBase,
   4]}]

(*Co associativity of o\[CapitalDelta]*)
Table[Subscript[o\[CapitalDelta], 2 -> 1, 2][
    Subscript[o\[CapitalDelta], 1 -> 2, 3][a]] -
   Subscript[o\[CapitalDelta], 2 -> 2, 3][
    Subscript[o\[CapitalDelta], 1 -> 1, 2][a]] // Expand, {a,
  Subscript[OBase, 1]}]

(*Unit comult & counit mult axioms*)
Table[Subscript[o\[Epsilon], 3][Subscript[om, 1, 2 -> 3][a*b]] -
  Subscript[o\[Epsilon], 1][a]*Subscript[o\[Epsilon], 2][b], {a,
  Subscript[OBase, 1]}, {b, Subscript[OBase, 2]}]
(Subscript[o\[CapitalDelta], 1 -> 2,
     3][(Subscript[G, 1] + Subscript[W, 1])] - (Subscript[G, 2] +
       Subscript[W, 2])*(Subscript[G, 3] + Subscript[W, 3])) // Expand

(*o\[CapitalDelta] is an alg hom*)
Table[Subscript[o\[CapitalDelta], 3 -> 4, 5][
    Subscript[om, 1, 2 -> 3][a*b]] -
   Subscript[om, 5, 7 -> 5][
```

```
    Subscript[om, 4, 6 -> 4][
     Subscript[o\[CapitalDelta], 2 -> 6, 7][
      Subscript[o\[CapitalDelta], 1 -> 4, 5][a*b]]]] // Expand, {a,
  Subscript[OBase, 1]}, {b, Subscript[OBase, 2]}]

(*Antipode axioms for oS*)(*note: 1 = Subscript[G, 1]+Subscript[W, 1].*)
Table[Subscript[om, 2, 3 -> 1][
   Subscript[oS, 2][Subscript[o\[CapitalDelta], 1 -> 2, 3][a]]] -
  Subscript[o\[Epsilon], 1][a], {a, Subscript[OBase, 1]}]
Table[Subscript[om, 2, 3 -> 1][
   Subscript[oS, 3][Subscript[o\[CapitalDelta], 1 -> 2, 3][a]]] -
  Subscript[o\[Epsilon], 1][a], {a, Subscript[OBase, 1]}]

(*Associativity of Hm*)(*Note: takes about an hour to run*)
Table[Subscript[Hm, 3, 4 -> 5][Subscript[Hm, 1, 2 -> 3][a*b*c]] -
  Subscript[Hm, 1, 3 -> 5][Subscript[Hm, 2, 4 -> 3][a*b*c]], {a,
  Subscript[DoubleBase, 1]}, {b, Subscript[DoubleBase, 2]}, {c,
  Subscript[DoubleBase, 4]}]

(*Statistical test to run*)
a = RandomChoice[Subscript[DoubleBase, 1]]
b = RandomChoice[Subscript[DoubleBase, 2]]
c = RandomChoice[Subscript[DoubleBase, 4]]
Subscript[Hm, 3, 4 -> 5][Subscript[Hm, 1, 2 -> 3][a*b*c]] -
 Subscript[Hm, 1, 3 -> 5][Subscript[Hm, 2, 4 -> 3][a*b*c]]
Clear[a, b, c]

(*The 0-2 move*)
Subscript[T, 1, 2]*Subscript[
\!\(\*OverscriptBox[\(T\), \(_\)]\), 3, 4] // Subscript[Hm, 2,
     4 -> 5] // Subscript[Hm, 5, 3 -> 6] // Subscript[Hm, 6,
  1 -> 7] // Simp

(*MP 1.3*)
LHS = Subscript[T, 31, 1]*Subscript[Tinv, 32, 2] // Subscript[Hm, 31,
  32 -> 3]
RHS1 := Subscript[Tinv, 121, 212]*Subscript[T, 3, 13] // Subscript[Hm,
      121, 13 -> 111] // Expand // SimpO
RHS2 := RHS1*Subscript[U, 10] // Subscript[Hm, 111, 10 -> 111] //
   Expand // SimpO
RHS3 := RHS2 *Subscript[T, 122, 211] // Subscript[Hm, 111,
     122 -> 111] // Expand // SimpO
RHS4 := RHS3*Subscript[U, 100] // Subscript[Hm, 111, 100 -> 1] //
     Expand // SimpO // Subscript[Hm, 211, 212 -> 2]
LHS - RHS4 // Expand // SimpO

(*MP 2.3 with \[CapitalOmega]0 applied to top vertex*)
LHS := Subscript[Tinv, 1, 31] Subscript[T, 2, 32] // Subscript[Hm, 31,
   32 -> 3]
RHS := Subscript[T, 212, 121] *Subscript[Tinv, 13, 3] *Subscript[
\!\(\*OverscriptBox[\(T\), \(_\)]\), 211, 122] // Subscript[Hm, 121,
     13 -> 100] // Subscript[Hm, 100, 122 -> 1] // Subscript[Hm, 211,
  212 -> 2]
LHS - RHS // SimpO

    (*MP 4.2*)
LHS = (Subscript[T, 1, 31] Subscript[T, 32, 2] // Subscript[Hm, 32,
     31 -> 3]);
RHS = (Subscript[T, 13, 31] Subscript[T, 12, 21] Subscript[T, 32,
         22] // Subscript[Hm, 13, 12 -> 1] // Subscript[Hm, 31,
       32 -> 3] // Subscript[Hm, 21, 22 -> 2]) ;
LHS - RHS // SimpO
```

```
(*Unimodular, and counimodular*)
Table[Subscript[Hm, 1, 2 -> 3][Subscript[uInt, 1]*a] -
  Subscript[u\[Epsilon], 2][a]*Subscript[uInt, 3], {a, Subscript[Base,
   2]}]
Table[Subscript[Hm, 2, 1 -> 3][a*Subscript[uInt, 1]] -
  Subscript[u\[Epsilon], 2][a]*Subscript[uInt, 3], {a, Subscript[Base,
   2]}]
Table[Subscript[Hm, 1, 2 -> 3][Subscript[oInt, 1]*a] -
  Subscript[o\[Epsilon], 2][a]*Subscript[oInt, 3], {a, Subscript[
  OBase, 2]}]
Table[Subscript[Hm, 2, 1 -> 3][a*Subscript[oInt, 1]] -
  Subscript[o\[Epsilon], 2][a]*Subscript[oInt, 3], {a, Subscript[
  OBase, 2]}]
Subscript[uS, 1][Subscript[uInt, 1]] - Subscript[uInt, 1]
Subscript[oS, 1][Subscript[oInt, 1]] - Subscript[oInt, 1]
Subscript[Pairing, 1, 2][(Subscript[uInt, 1])*Subscript[oInt, 2]]

(*Pivot*)
Table[Subscript[um, 1, 2 -> 3][Subscript[u, 1]*a] -
  Subscript[um, 2, 1 -> 3][
   Subscript[uS, 2][Subscript[uS, 2][Subscript[u, 1]*a]]], {a,
  Subscript[Base, 2]}]
Subscript[u\[CapitalDelta], 1 -> 2, 3][Subscript[u, 1]] -
 Subscript[u, 2] Subscript[u, 3]
Subscript[u\[Epsilon], 1][Subscript[u, 1]]

    (*Dualpivot*)
Table[Subscript[om, 1, 2 -> 3][Subscript[U, 1]*a] -
  Subscript[om, 2, 1 -> 3][
   Subscript[oS, 2][Subscript[oS, 2][Subscript[U, 1]*a]]], {a,
  Subscript[Base, 2]}]
Subscript[o\[CapitalDelta], 1 -> 2, 3][Subscript[U, 1]] -
  Subscript[U, 2] Subscript[U, 3] // Expand
Subscript[o\[Epsilon], 1][Subscript[U, 1]]

(*check \[Mu](abu)=\[Mu](bau), \[Mu] in H^*. i.e. abuse of notation*)
LHS = Table[
  Subscript[Pairing, 10, 5][
   Subscript[Hm, 4, 1 -> 5][
    Subscript[Hm, 2, 3 -> 4][
     a*b*Subscript[U, 1]*(Subscript[uInt, 10])]]], {a, Subscript[
   OBase, 2]}, {b, Subscript[OBase, 3]}]
RHS = Table[
  Subscript[Pairing, 10, 5][
   Subscript[Hm, 4, 1 -> 5][
    Subscript[Hm, 3, 2 -> 4][
     b*a*Subscript[U, 1]*(Subscript[uInt, 10])]]], {a, Subscript[
   OBase, 2]}, {b, Subscript[OBase, 3]}]
LHS - RHS

(*Radford's identity with \[Alpha]=1*)
Table[Subscript[Pairing, 8, 6][
   Subscript[um, 7, 2 -> 8][
     Subscript[um, 5, 4 -> 7][
      Subscript[um, 3, 1 -> 5][Subscript[u, 3]*a*Subscript[u, 4]*b]]]*
    Subscript[XY, 6]] -
  Subscript[Pairing, 4, 6][
   Subscript[um, 2, 1 -> 4][b*a]*Subscript[XY, 6]], {a, Subscript[
  Base, 1]}, {b, Subscript[Base, 2]}]

(*Check if FockMatix is a representation*)
```

```
Table[Subscript[FockMatrix, 1][a] . Subscript[FockMatrix, 2][b] -
  Subscript[FockMatrix, 3][Subscript[Hm, 1, 2 -> 3][a*b]], {a,
  Subscript[DoubleBase, 1]}, {b, Subscript[DoubleBase, 2]}]

(*Fock of the loop. It agrees*)
Subscript[FockMatrix, 1][
  Subscript[Hm, 5, 3 -> 1][
   Subscript[Hm, 2, 4 -> 5][
    Subscript[T, 2, 3]*Subscript[U, 4]]]] // MatrixForm
Table[Subscript[Pairing, 4, 3][
   Subscript[Hm, 2, 1 -> 3][Subscript[U, 2]*a*Subscript[uInt, 4]]]*
  Subscript[XY, 1], {a, Subscript[OBase, 1]}]
```

10.3 Initialization cell image form

```
(*Note: never use x or y as variable names.*)
Base_i_ := {1, g_i, x_i, y_i, gx_i, gy_i, xy_i, gxy_i}
SimpO[Z_] := Z /. {a_ * G_i_ :> a * (1 - W_i), G_i_ :> (1 - W_i)} // Expand
```

```
um_{i_,j_→k_}[Z_] := Expand[Z] /. {
    g_i g_j → 1,
    g_i x_j → gx_k,
    g_i y_j → gy_k,
    g_i gx_j → x_k,
    g_i gy_j → y_k,
    g_i xy_j → gxy_k,
    g_i gxy_j → xy_k,
    x_i g_j → g_k - 1 - gx_k,
    x_i x_j → x_k,
    x_i y_j → xy_k,
    x_i gx_j → -x_k,
    x_i gy_j → gy_k - y_k - gxy_k,
    x_i xy_j → xy_k,
    x_i gxy_j → -xy_k,
    y_i g_j → g_k - 1 - gy_k,
    y_i x_j → x_k + y_k - xy_k,
    y_i y_j → y_k,
    y_i gx_j → -x_k - gy_k + gxy_k,
```

```
  y_i gy_j → -y_k,
  y_i xy_j → y_k,
  y_i gxy_j → gxy_k - xy_k - gy_k,
  gx_i g_j → 1 - g_k - x_k,
  gx_i x_j → gx_k,
  gx_i y_j → gxy_k,
  gx_i gx_j → -gx_k,
  gx_i gy_j → y_k - gy_k - xy_k,
  gx_i xy_j → gxy_k,
  gx_i gxy_j → -gxy_k,
  gy_i g_j → 1 - g_k - y_k,
  gy_i x_j → gx_k + gy_k - gxy_k,
  gy_i y_j → gy_k,
  gy_i gx_j → -gx_k - y_k + xy_k,
  gy_i gy_j → -gy_k,
  gy_i xy_j → gy_k,
  gy_i gxy_j → -gxy_k - y_k + xy_k,
  xy_i g_j → gxy_k + y_k - gy_k - x_k - gx_k - 1 + g_k,
  xy_i x_j → x_k,
  xy_i y_j → xy_k,
  xy_i gx_j → -x_k - gy_k + y_k - xy_k + gxy_k,
  xy_i gy_j → -xy_k,
  xy_i xy_j → xy_k,
  xy_i gxy_j → -2 xy_k - gy_k + y_k + gxy_k,
  gxy_i g_j → xy_k + gy_k - y_k - gx_k - x_k - g_k + 1,
  gxy_i x_j → gx_k,
  gxy_i y_j → gxy_k,
  gxy_i gx_j → -gx_k - y_k + gy_k - gxy_k + xy_k,
  gxy_i gy_j → -gxy_k,
  gxy_i xy_j → gxy_k,
  gxy_i gxy_j → -2 gxy_k - y_k + gy_k + xy_k
} /. {x_i :→ x_k, x_j :→ x_k} // Expand
```

```
uΔi_→j_,k_[Z_] := Expand[Z] /. {
    gi → gj gk,
    xi → xj gk + xk,
    yi → yj gk + yk,
    gxi → gxj + gj gxk,
    gyi → gyj + gj gyk,
    xyi → xyj + xj gyk − yj + yj gk − yj gxk + xyk,
    gxyi → gxyj gk + gxj yk − gyj gk + gyj − gyj xk + gj gxyk
  } // Expand

uϵi_[Z_] := Expand[Z] /. {
   gi → 1,
   xi → 0,
   yi → 0,
   gxi → 0,
   gyi → 0,
   xyi → 0,
   gxyi → 0
  }
```

```
uS_i_ [Z_] := Expand[Z] /. {
    g_i → g_i,
    x_i → 1 - g_i + gx_i,
    y_i → 1 - g_i + gy_i,
    gx_i → -x_i,
    gy_i → -y_i,
    xy_i → 1 - g_i + gy_i - x_i + xy_i,
    gxy_i → gxy_i - gy_i - x_i
  } // Expand

uS̄_i_ [Z_] := Expand[Z] /. {
    g_i → g_i,
    x_i → -gx_i,
    y_i → -gy_i,
    gx_i → -1 + g_i + x_i,
    gy_i → -1 + g_i + y_i,
    xy_i → -gx_i + xy_i - y_i,
    gxy_i → -1 + g_i - gx_i + gxy_i + y_i
  } // Expand
```

```
(*Dual*)
OBase_i_ := {W_i, G_i, X_i, Y_i, GX_i, GY_i, XY_i, GXY_i}

Pairing_i_,j_ [Z_] := Expand[Z] /. {
  g_i W_j → 0, g_i G_j → 1, g_i X_j → 0, g_i Y_j → 0, g_i GX_j → 0, g_i GY_j → 0, g_i XY_j → 0, g_i GXY_j → 0, x_i W_j → 0, x_i G_j → 0,
  x_i X_j → 1, x_i Y_j → 0, x_i GX_j → 0, x_i GY_j → 0, x_i XY_j → 0, x_i GXY_j → 0, y_i W_j → 0, y_i G_j → 0, y_i X_j → 0, y_i Y_j → 1,
  y_i GX_j → 0, y_i GY_j → 0, y_i XY_j → 0, y_i GXY_j → 0, gx_i W_j → 0, gx_i G_j → 0, gx_i X_j → 0, gx_i Y_j → 0, gx_i GX_j → 1,
  gx_i GY_j → 0, gx_i XY_j → 0, gx_i GXY_j → 0, gy_i W_j → 0, gy_i G_j → 0, gy_i X_j → 0, gy_i Y_j → 0, gy_i GX_j → 0, gy_i GY_j → 1,
  gy_i XY_j → 0, gy_i GXY_j → 0, xy_i W_j → 0, xy_i G_j → 0, xy_i X_j → 0, xy_i Y_j → 0, xy_i GX_j → 0, xy_i GY_j → 0, xy_i XY_j → 1,
  xy_i GXY_j → 0, gxy_i W_j → 0, gxy_i G_j → 0, gxy_i X_j → 0, gxy_i Y_j → 0, gxy_i GX_j → 0, gxy_i GY_j → 0, gxy_i XY_j → 0,
  gxy_i GXY_j → 1, W_j → 1, G_j → 0, X_j → 0, Y_j → 0, GX_j → 0, GY_j → 0, XY_j → 0, GXY_j → 0, g_i → 1, x_i → 0, y_i → 0,
  gx_i → 0, gy_i → 0, xy_i → 0, gxy_i → 0}
```

```
om_i_,j_→k_ [Z_] := Expand[Z] /. {
    W_i W_j → W_k, G_i W_j → 0, X_i W_j → 0, Y_i W_j → -XY_k, GX_i W_j → GX_k, GY_i W_j → GXY_k + GY_k, XY_i W_j → XY_k, GXY_i W_j → 0, W_i G_j → 0,
    G_i G_j → G_k, X_i G_j → X_k, Y_i G_j → XY_k + Y_k, GX_i G_j → 0, GY_i G_j → -GXY_k, XY_i G_j → 0, GXY_i G_j → GXY_k, W_i X_j → X_k, G_i X_j → 0,
    X_i X_j → 0, Y_i X_j → 0, GX_i X_j → 0, GY_i X_j → -GXY_k, XY_i X_j → 0, GXY_i X_j → 0, W_i Y_j → Y_k, G_i Y_j → 0, X_i Y_j → 0, Y_i Y_j → 0,
    GX_i Y_j → GXY_k, GY_i Y_j → 0, XY_i Y_j → 0, GXY_i Y_j → 0, W_i GX_j → 0, G_i GX_j → GX_k, X_i GX_j → 0, Y_i GX_j → -XY_k, GX_i GX_j → 0,
    GY_i GX_j → 0, XY_i GX_j → 0, GXY_i GX_j → 0, W_i GY_j → 0, G_i GY_j → GY_k, X_i GY_j → XY_k, Y_i GY_j → 0, GX_i GY_j → 0, GY_i GY_j → 0,
    XY_i GY_j → 0, GXY_i GY_j → 0, W_i XY_j → XY_k, G_i XY_j → 0, X_i XY_j → 0, Y_i XY_j → 0, GX_i XY_j → 0, GY_i XY_j → 0, XY_i XY_j → 0,
    GXY_i XY_j → 0, W_i GXY_j → 0, G_i GXY_j → GXY_k, X_i GXY_j → 0, Y_i GXY_j → 0, GX_i GXY_j → 0, GY_i GXY_j → 0, XY_i GXY_j → 0, GXY_i GXY_j → 0} /.
  {a_i_ ⧴ a_k, a_j_ ⧴ a_k} // Expand
```

```
oΔ_{i_→k_,j_}[Z_] := Expand[Z] /. {
  W_i → W_j W_k + G_j G_k + G_j GX_k + G_j GXY_k + G_j GY_k − G_j X_k − G_j XY_k − G_j Y_k,
  G_i → G_j W_k + W_j G_k − G_j GX_k − G_j GXY_k − G_j GY_k + G_j X_k + G_j XY_k + G_j Y_k,
  X_i → G_k GX_j − G_j GX_k − G_j GXY_k + X_j W_k + W_j X_k − GX_j X_k + X_j X_k − G_j XY_k − GX_j XY_k + X_j XY_k − GX_j Y_k + X_j Y_k,
  Y_i → −G_j GXY_k − GX_j GXY_k − GXY_j GXY_k + G_k GY_j + GX_k GY_j − G_j GY_k − GX_j GY_k − GXY_j GY_k − GY_j X_k + G_j XY_k + GX_j XY_k + GXY_j XY_k +
    Y_j W_k + W_j Y_k − GY_j Y_k + X_j Y_k + XY_j Y_k + Y_j Y_k,
  GX_i → GX_j W_k + W_j GX_k − GX_j GX_k − G_j GXY_k − GX_j GXY_k − GX_j GY_k + G_k X_j + GX_k X_j + GXY_k X_j + GY_k X_j − G_j X_k − G_j XY_k,
  GY_i → G_j GXY_k + GX_j GXY_k + GXY_j GXY_k + GY_j W_k − GX_k GY_j + W_j GY_k − GY_j GY_k + GY_k X_j + GY_j X_k + GY_k XY_j − G_j XY_k − GX_j XY_k −
    GXY_j XY_k + G_k Y_j + GY_k Y_j − G_j Y_k − GX_j Y_k − GXY_j Y_k,
  XY_i → G_k GXY_j + G_j GXY_k + GX_j GXY_k + GXY_j GXY_k − GX_k GY_j + GX_j GY_k + GXY_j GY_k − GXY_j X_k + XY_j W_k + X_k XY_j + W_j XY_k − GX_j XY_k −
    2 * GXY_j XY_k − GY_j XY_k + XY_j XY_k + X_k Y_j + XY_k Y_j − GXY_j Y_k − X_j Y_k,
  GXY_i → GXY_j W_k − GX_k GXY_j + W_j GXY_k − GX_j GXY_k − 2 * GXY_j GXY_k − GXY_k GY_j − GXY_j GY_k − GY_k X_j − GY_j X_k + G_k XY_j + GX_k XY_j +
    GXY_k XY_j + G_j XY_k + GX_j XY_k + GXY_j XY_k + GX_k Y_j + GXY_k Y_j + GX_j Y_k + GXY_j Y_k
 }
```

```
oϵ_{i_}[Z_] := Expand[Z] /. {
  W_i → 1,
  G_i → 0,
  X_i → 0,
  Y_i → 0,
  GX_i → 0,
  GY_i → 0,
  XY_i → 0,
  GXY_i → 0
 }
oS_{i_}[Z_] := Expand[Z] /. {
  W_i → W_i + X_i + Y_i + XY_i,
  G_i → G_i − X_i − Y_i − XY_i,
  X_i → −GX_i − XY_i − GXY_i,
  Y_i → −GY_i,
  GX_i → X_i,
  GY_i → Y_i + XY_i − GXY_i,
  XY_i → XY_i,
  GXY_i → GXY_i
 }
```

```
oS̄i_[Z_] := Expand[Z] /. {
    Wi → -GXi - GXYi - GYi + Wi,
    Gi → Gi + GXi + GXYi + GYi,
    Xi → GXi,
    Yi → GXYi + GYi - XYi,
    GXi → -GXYi - Xi - XYi,
    GYi → -Yi,
    XYi → XYi,
    GXYi → GXYi}

OCompi_,j_[Z_] := Expand[Z] /. {a_i a_j :> 1, b_i b_j :> 1, a_i b_j :> 0}

FockMatrixi_[c_] := Module[{o1, u1, cnew, j, k, m, n, o},
   cnew = SplitOUi,o1,u1[c];
   Table[OCompn,o[b * z], {b, Table[Hmk,o1→n[Pairingu1,m[oΔj→m,k[cnew * a]]], {a , OBasej}]}, {z, OBaseo}]
  ]

(*Heisenberg double*)

SplitOUi_,j_,k_[Z_] := Expand[Z] /. {
    gi → gk, xi → xk, yi → yk, gxi → gxk, gyi → gyk, xyi → xyk, gxyi → gxyk,
    Wi → Wj, Gi → Gj, Xi → Xj, Yi → Yj, GXi → GXj, GYi → GYj, XYi → XYj, GXYi → GXYj
   }

Hmi_,j_→k_[Z_] := Module[{i1, i2, j1, j2, i11, i12, j21, j22, Znew},
   Znew = SplitOUj,j1,j2[SplitOUi,i1,i2[Expand[Z]]];
   Expand[Znew] // oΔi1→i11,i12 // uΔj2→j21,j22 // Pairingj22,i11 // umi2,j21→k // omi12,j1→k // Expand
  ] (*I would use this, then you don't to add Wj+Gj in every expression solely involing gi's*)

DoubleBasei_ := Table[a * b, {a, Basei}, {b, OBasei}] // Flatten
Ti_,j_ := Total[OBasei * Basej]
T̄i_,j_ := Total[OBasei * uS̄j[ Basej]]
Tinvi_,j_ := Total[OBasei * uSj[ Basej]]

Ui_ := -Gi - GXi - GXYi - GYi + Wi + Xi + XYi + Yi
ui_ := gi
uinvi_ := gi
Uinvi_ := -Gi - GXi - GXYi - GYi + Wi + Xi + XYi + Yi
uInti_ := xyi + gxyi - xi - gxi
oInti_ := XYi
```

10.4 Other lines to run image form

In[∘]:=

```
(*Pairing axiom*)
Table[Pairing_{1,2}[a*b], {a, Base_1}, {b, OBase_2}] // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

In[∘]:=

```
(*Multiplication table*)
Table[um_{1,2→3}[a*b], {a, Base_1}, {b, Base_2}] // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & g_3 & x_3 & y_3 & gx_3 & gy_3 & xy_3 & gxy_3 \\ g_3 & 1 & gx_3 & gy_3 & x_3 & y_3 & gxy_3 & xy_3 \\ x_3 & -1+g_3-gx_3 & x_3 & xy_3 & -x_3 & -gxy_3+gy_3-y_3 & xy_3 & -xy_3 \\ y_3 & -1+g_3-gy_3 & x_3-xy_3+y_3 & y_3 & gxy_3-gy_3-x_3 & -y_3 & y_3 & gxy_3-gy_3-xy_3 \\ gx_3 & 1-g_3-x_3 & gx_3 & gxy_3 & -gx_3 & -gy_3-xy_3+y_3 & gxy_3 & -gxy_3 \\ gy_3 & 1-g_3-y_3 & gx_3-gxy_3+gy_3 & gy_3 & -gx_3+xy_3-y_3 & -gy_3 & gy_3 & -gxy_3+xy_3-y_3 \\ xy_3 & -1+g_3-gx_3+gxy_3-gy_3-x_3+y_3 & x_3 & xy_3 & gxy_3-gy_3-x_3-xy_3+y_3 & -xy_3 & xy_3 & gxy_3-gy_3-2\,xy_3+y_3 \\ gxy_3 & 1-g_3-gx_3+gy_3-x_3+xy_3-y_3 & gx_3 & gxy_3 & -gx_3-gxy_3+gy_3+xy_3-y_3 & -gxy_3 & gxy_3 & -2\,gxy_3+gy_3+xy_3-y_3 \end{pmatrix}$$

In[∘]:=

```
(*Antipode axioms*)
Table[um_{2,3→1}[uS_2[uΔ_{1→2,3}[a]]] - uϵ_1[a], {a, Base_1}]
Table[um_{2,3→1}[uS_3[uΔ_{1→2,3}[a]]] - uϵ_1[a], {a, Base_1}]
```

Out[∘]= {0, 0, 0, 0, 0, 0, 0, 0}

Out[∘]= {0, 0, 0, 0, 0, 0, 0, 0}

In[∘]:=

```
(*uΔ is an algebra morphism*)
Table[uΔ_{3→4,5}[um_{1,2→3}[a*b]] - um_{5,7→5}[um_{4,6→4}[uΔ_{2→6,7}[uΔ_{1→4,5}[a*b]]]], {a, Base_1}, {b, Base_2}]
```

Out[∘]= {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}

In[∘]:=

```
(* co-unit + product (the other one is automatically satisfied)*)
Table[uϵ_3[um_{1,2→3}[a*b]] - uϵ_1[a]*uϵ_2[b], {a, Base_1}, {b, Base_2}]
(uΔ_{1→2,3}[1] - 1*1) // Expand
```

Out[∘]= {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}

```
In[•]:= (*co-associativity of uΔ*)
        Table[uΔ_{2→1,2}[uΔ_{1→2,3}[a]] - uΔ_{2→2,3}[uΔ_{1→1,2}[a]], {a, Base_1}]

Out[•]= {0, 0, 0, 0, 0, 0, 0, 0}
```

```
(*Associativity of um*)
Table[um_{3,4→5}[um_{1,2→3}[a*b*c]] - um_{1,3→5}[um_{2,4→3}[a*b*c]], {a, Base_1}, {b, Base_2}, {c, Base_4}]

Out[•]= {{{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
  {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
  {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
  {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
  {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
  {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
  {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
  {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}}
```

```
(*Antipode has order 4, and not 2.*)
Table[uS_1[uS_1[uS_1[uS_1[a]]]] - a, {a, Base_1}]
Table[uS_1[uS_1[a]] - a, {a, Base_1}]

Out[•]= {0, 0, 0, 0, 0, 0, 0, 0}

Out[•]= {0, 0, 1 - g_1 - 2 x_1, 1 - g_1 - 2 y_1, -1 + g_1 - 2 gx_1, -1 + g_1 - 2 gy_1, 1 - g_1 - gx_1 + gy_1 - x_1 - y_1, -1 + g_1 - gx_1 - gy_1 - x_1 + y_1}
```

```
In[•]:=
   (*Unit in Om*)
   Table[om_{2,1→2}[(W_1 + G_1) * a] - a, {a, OBase_2}]

Out[•]= {0, 0, 0, 0, 0, 0, 0, 0}
```

```
In[∘]:= (*Associativity of om*)
        Table[om_{3,4→5}[om_{1,2→3}[a * b * c]] - om_{1,3→5}[om_{2,4→3}[a * b * c]], {a, OBase_1}, {b, OBase_2}, {c, OBase_4}]

Out[∘]= {{{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
          {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
         {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
          {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
         {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
          {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
         {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
          {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
         {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
          {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
         {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
          {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
         {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
          {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
         {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
          {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}}

In[∘]:= (*Co associativity of oΔ*)
        Table[oΔ_{2→1,2}[oΔ_{1→2,3}[a]] - oΔ_{2→2,3}[oΔ_{1→1,2}[a]] // Expand, {a, OBase_1}]

Out[∘]= {0, 0, 0, 0, 0, 0, 0, 0}

In[∘]:= (*Unit comult & counit mult axioms*)
        Table[oϵ_3[om_{1,2→3}[a * b]] - oϵ_1[a] * oϵ_2[b], {a, OBase_1}, {b, OBase_2}]
        (oΔ_{1→2,3}[(G_1 + W_1)] - (G_2 + W_2) * (G_3 + W_3)) // Expand

Out[∘]= {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
         {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}

Out[∘]= 0

In[∘]:= (*oΔ is an alg hom*)
        Table[oΔ_{3→4,5}[om_{1,2→3}[a * b]] - om_{5,7→5}[om_{4,6→4}[oΔ_{2→6,7}[oΔ_{1→4,5}[a * b]]]] // Expand, {a, OBase_1}, {b, OBase_2}]

Out[∘]= {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
         {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}

        (*Antipode axioms for oS*) (*note: 1 = G_1+W_1.*)
        Table[om_{2,3→1}[oS_2[oΔ_{1→2,3}[a]]] - oϵ_1[a], {a, OBase_1}]
        Table[om_{2,3→1}[oS_3[oΔ_{1→2,3}[a]]] - oϵ_1[a], {a, OBase_1}]

Out[∘]= {-1 + G_1 + W_1, 0, 0, 0, 0, 0, 0, 0}

Out[∘]= {-1 + G_1 + W_1, 0, 0, 0, 0, 0, 0, 0}
```

In[∘]:=

```
(*Associativity of Hm*)(*Note: takes about an hour to run*)
Table[Hm_{3,4→5}[Hm_{1,2→3}[a * b * c]] - Hm_{1,3→5}[Hm_{2,4→3}[a * b * c]], {a, DoubleBase_1}, {b, DoubleBase_2}, {c, DoubleBase_4}]
```

Out[∘]=

```
{{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, <<47>>,
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}},
 {<<1>>}, {<<1>>}, {<<1>>}, <<56>>, {<<1>>}, {<<1>>}, {<<1>>}, {<<1>>}}
```

Full expression not available (original memory size: 6.6 MB)

In[∘]:=

```
(*Statistical test to run*)
a = RandomChoice[DoubleBase_1]
b = RandomChoice[DoubleBase_2]
c = RandomChoice[DoubleBase_4]
Hm_{3,4→5}[Hm_{1,2→3}[a * b * c]] - Hm_{1,3→5}[Hm_{2,4→3}[a * b * c]]
Clear[a, b, c]
```

Out[∘]= $gx_1\ GY_1$

Out[∘]= $gy_2\ X_2$

Out[∘]= $gxy_4\ W_4$

Out[∘]= 0

In[∘]:=

```
(*The 0-2 move*)
T_{1,2} * T̄_{3,4} // Hm_{2,4→5} // Hm_{5,3→6} // Hm_{6,1→7} // Simp0
```

Out[∘]= 1

In[34]:=

```
(*MP 1.3*)
LHS = T_{31,1} * Tinv_{32,2} // Hm_{31,32→3}
RHS1 := Tinv_{121,212} * T_{3,13} // Hm_{121,13→111} // Expand // Simp0
RHS2 := RHS1 * U_{10} // Hm_{111,10→111} // Expand // Simp0
RHS3 := RHS2 * T_{122,211} // Hm_{111,122→111} // Expand // Simp0
RHS4 := RHS3 * U_{100} // Hm_{111,100→1} // Expand // Simp0 // Hm_{211,212→2}
LHS - RHS4 // Expand // Simp0
```

Out[34]= $g_1\, g_2\, G_3 + gx_1\, GX_3 + gx_1\, GXY_3 - g_2\, gx_1\, GXY_3 + g_2\, gxy_1\, GXY_3 + g_1\, gxy_2\, GXY_3 - gx_2\, GXY_3\, gy_1 - g_1\, GXY_3\, gy_2 + gx_1\, GXY_3\, gy_2 + gy_1\, GY_3 + W_3 - g_1\, GX_3\, x_2 - g_1\, GXY_3\, x_2 + X_3 - g_2\, X_3 + gx_2\, X_3 + g_2\, x_1\, X_3 + XY_3 - g_2\, XY_3 + gy_2\, XY_3 - x_2\, XY_3 + xy_1\, XY_3 + xy_2\, XY_3 - XY_3\, y_1 + g_2\, XY_3\, y_1 + x_2\, XY_3\, y_1 - g_1\, GY_3\, y_2 - x_1\, XY_3\, y_2 + Y_3 - g_2\, Y_3 + gy_2\, Y_3 + g_2\, y_1\, Y_3$

Out[39]= 0

```
In[72]:= (*MP 2.3 with Ω0 applied to top vertex*)
        LHS := Tinv_{1,31} T_{2,32} // Hm_{31,32→3}
        RHS := T_{212,121} * Tinv_{13,3} * T̄_{211,122} // Hm_{121,13→100} // Hm_{100,122→1} // Hm_{211,212→2}
        LHS - RHS // SimpO

Out[74]= 0

In[ ]:=
       (*MP 4.2*)
       LHS = (T_{1,31} T_{32,2} // Hm_{32,31→3});
       RHS = (T_{13,31} T_{12,21} T_{32,22} // Hm_{13,12→1} // Hm_{31,32→3} // Hm_{21,22→2});
       LHS - RHS // SimpO

Out[ ]= 0

In[ ]:= (*Unimodular, and counimodular*)
        Table[Hm_{1,2→3}[uInt_1 * a] - uϵ_2[a] * uInt_3, {a, Base_2}]
        Table[Hm_{2,1→3}[a * uInt_1] - uϵ_2[a] * uInt_3, {a, Base_2}]
        Table[Hm_{1,2→3}[oInt_1 * a] - oϵ_2[a] * oInt_3, {a, OBase_2}]
        Table[Hm_{2,1→3}[a * oInt_1] - oϵ_2[a] * oInt_3, {a, OBase_2}]
        uS_1[uInt_1] - uInt_1
        oS_1[oInt_1] - oInt_1
        Pairing_{1,2}[(uInt_1) * oInt_2]

Out[ ]= {0, 0, 0, 0, 0, 0, 0, 0}

Out[ ]= {0, 0, 0, 0, 0, 0, 0, 0}

Out[ ]= {0, 0, 0, 0, 0, 0, 0, 0}

Out[ ]= {0, 0, 0, 0, 0, 0, 0, 0}

Out[ ]= 0

Out[ ]= 0

Out[ ]= 1
```

```
In[ ]:=
(*Pivot*)
Table[um1,2→3[u1 * a] - um2,1→3[uS2[uS2[u1 * a]]], {a, Base2}]
uΔ1→2,3[u1] - u2 u3
uϵ1[u1]
Out[ ]= {0, 0, 0, 0, 0, 0, 0, 0}
Out[ ]= 0
Out[ ]= 1

In[ ]:=
(*Pivot*)
Table[um1,2→3[u1 * a] - um2,1→3[uS2[uS2[u1 * a]]], {a, Base2}]
uΔ1→2,3[u1] - u2 u3
uϵ1[u1]
Out[ ]= {0, 0, 0, 0, 0, 0, 0, 0}
Out[ ]= 0
Out[ ]= 1

(*Dualpivot*)
Table[om1,2→3[U1 * a] - om2,1→3[oS2[oS2[U1 * a]]], {a, Base2}]
oΔ1→2,3[U1] - U2 U3 // Expand
oϵ1[U1]
Out[ ]= {0, 0, 0, 0, 0, 0, 0, 0}
Out[ ]= 0
Out[ ]= 1
```

```
(*check μ(abu)=μ(bau), μ in H*. i.e. abuse of notation*)
LHS = Table[Pairing_{5,10}[Hm_{4,1→5}[Hm_{2,3→4}[a*b*u_1*(XY_10)]]], {a, Base_2}, {b, Base_3}]
RHS = Table[Pairing_{5,10}[Hm_{4,1→5}[Hm_{3,2→4}[b*a*u_1*(XY_10)]]], {a, Base_2}, {b, Base_3}]
LHS - RHS

LHS = Table[Pairing_{10,5}[Hm_{4,1→5}[Hm_{2,3→4}[a*b*U_1*(uInt_10)]]], {a, OBase_2}, {b, OBase_3}]
RHS = Table[Pairing_{10,5}[Hm_{4,1→5}[Hm_{3,2→4}[b*a*U_1*(uInt_10)]]], {a, OBase_2}, {b, OBase_3}]
LHS - RHS
```

```
Out[∘]= {{0, 0, 0, 0, 0, 0, 0, 1}, {0, 0, 0, 0, 0, 0, 1, 0}, {0, 0, 0, 0, 0, -1, 0, 0}, {0, 0, 0, 0, 1, 0, 0, 1},
  {0, 0, 0, 1, 0, 0, 1, -1}, {0, 0, -1, 0, 0, 0, 0, -1}, {0, 1, 0, 0, 1, 0, 0, 1}, {1, 0, 0, 1, -1, -1, 1, -2}}

Out[∘]= {{0, 0, 0, 0, 0, 0, 0, 1}, {0, 0, 0, 0, 0, 0, 1, 0}, {0, 0, 0, 0, 0, -1, 0, 0}, {0, 0, 0, 0, 1, 0, 0, 1},
  {0, 0, 0, 1, 0, 0, 1, -1}, {0, 0, -1, 0, 0, 0, 0, -1}, {0, 1, 0, 0, 1, 0, 0, 1}, {1, 0, 0, 1, -1, -1, 1, -2}}

Out[∘]= {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}

Out[∘]= {{0, 0, 0, -1, 0, 0, 1, 0}, {0, 0, 0, 0, 0, 1, 0, -1}, {0, 0, 0, 0, 0, 1, 0, 0}, {-1, 0, 0, 0, -1, 0, 0, 0},
  {0, 0, 0, -1, 0, 0, 0, 0}, {0, 1, 1, 0, 0, 0, 0, 0}, {1, 0, 0, 0, 0, 0, 0, 0}, {0, -1, 0, 0, 0, 0, 0, 0}}

Out[∘]= {{0, 0, 0, -1, 0, 0, 1, 0}, {0, 0, 0, 0, 0, 1, 0, -1}, {0, 0, 0, 0, 0, 1, 0, 0}, {-1, 0, 0, 0, -1, 0, 0, 0},
  {0, 0, 0, -1, 0, 0, 0, 0}, {0, 1, 1, 0, 0, 0, 0, 0}, {1, 0, 0, 0, 0, 0, 0, 0}, {0, -1, 0, 0, 0, 0, 0, 0}}

Out[∘]= {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}
```

```
In[∘]:= (*Radford's identity with α=1*)
Table[Pairing_{8,6}[um_{7,2→8}[um_{5,4→7}[um_{3,1→5}[u_3*a*u_4*b]]]*XY_6] - Pairing_{4,6}[um_{2,1→4}[b*a]*XY_6], {a, Base_1}, {b, Base_2}]

Out[∘]= {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}
```

```
(*Check if FockMatix is a representation*)
Table[FockMatrix_1[a].FockMatrix_2[b] - FockMatrix_3[Hm_{1,2→3}[a*b]], {a, DoubleBase_1}, {b, DoubleBase_2}]

Out[∘]= {{{{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
   {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}},
  {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
   {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}, … 61 …, {{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
   {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}}, … 62 …,
 {{{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, … 4 …, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}, … 62 …, {… 1 …}}}
Full expression not available (original memory size: 2.6 MB)
```

```
(*Fock of the loop. It agrees*)
FockMatrix_1[Hm_{5,3→1}[Hm_{2,4→5}[T_{2,3}*U_4]]] // MatrixForm
Table[Pairing_{4,3}[Hm_{2,1→3}[U_2*a*uInt_4]]*XY_1, {a, OBase_1}]
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$

```
Out[∘]= {0, 0, 0, -XY_1, 0, XY_1, XY_1, -XY_1}
```