

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

A finite presentation of knotted trivalent graphs

Permalink

<https://escholarship.org/uc/item/8x6589zw>

Author

Comstock, Jana

Publication Date

2008

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

A Finite Presentation of Knotted Trivalent Graphs

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics

by

Jana Comstock

Committee in charge:

Professor Justin Roberts, Chair
Professor Peter Teichner, Co-Chair
Professor Benjamin Grinstein
Professor Kenneth Intriligator
Professor David Meyer

2008

Copyright
Jana Comstock, 2008
All rights reserved.

The dissertation of Jana Comstock is approved,
and it is acceptable in quality and form for publi-
cation on microfilm:

Co-Chair

Chair

University of California, San Diego

2008

DEDICATION

To my parents, for their unwavering support of my consistently unprofitable career choices.

TABLE OF CONTENTS

	Signature Page	iii
	Dedication	iv
	Table of Contents	v
	List of Figures	vi
	Acknowledgements	viii
	Vita and Publications	ix
	Abstract	x
Chapter 1	Introduction	1
Chapter 2	Background	4
	2.1 Reidemeister moves	4
	2.2 The Kontsevich Integral	5
	2.3 TG Algebras	6
	2.4 The Århus Integral and KTG manifold invariants	7
Chapter 3	Knotted trivalent graph diagrams	8
	3.1 Sets	8
	3.2 Operations	11
	3.3 Generators	21
	3.4 Relations	25
Chapter 4	Knotted trivalent graphs	34
	4.1 Sets	34
	4.2 Operations	35
	4.3 Generators	37
	4.4 Relations	38
Appendix A	The pentagon relation	44
Bibliography	46

LIST OF FIGURES

Figure 3.1: Two KTGDs which represent different KTGs in 3-space	9
Figure 3.2: Two KTGEs which represent the same KTGD with preferred edge indicated in green	10
Figure 3.3: Two KTGDs which represent the same KTGD with maximal tree indicated in red	10
Figure 3.4: Maps between sets	11
Figure 3.5: Connect sum	12
Figure 3.6: Unzip	12
Figure 3.7: Alternate connect sum	13
Figure 3.8: Switch leaf for a trivalent vertex	14
Figure 3.9: Switch branch	14
Figure 3.10: Maps between sets	15
Figure 3.11: Connectsum	16
Figure 3.12: An example of unzip	16
Figure 3.13: Kill	17
Figure 3.14: Bubble	18
Figure 3.15: Vertex connect sum	18
Figure 3.16: Tree connect sum	19
Figure 3.17: The disjoint union in the top line is undone by the sequence of unzips and kills below.	19
Figure 3.18: The connect sum in the top line is undone by the connect sum, sequence of unzips, and disjoint union below.	20
Figure 3.19: Inverting an appropriate unzip	21
Figure 3.20: A possible blueprint (step 0) for	22
Figure 3.21: The completion of step 1 for	23
Figure 3.22: Step 2	23
Figure 3.23: Step 3	24
Figure 3.24: Tetrahedron relation	25
Figure 3.25: Twisted tetrahedron relation	26
Figure 3.26: Connect sum relation	27
Figure 3.27: Unzip relation	28
Figure 3.28: A commutative triangle reconciling different formulations of the unzip relation	28
Figure 3.29: A composition of relations	29
Figure 3.30: A relation showing that the algorithm on	30
Figure 3.31: The composition of relations corresponding to a sequence of operations which generates	31
Figure 3.32: The composition of relations corresponding to another sequence of operations which generates	32

Figure 3.33: The relation obtained by noticing that both sequences above are equivalent to the algorithm	33
Figure 4.1: Maps between sets	35
Figure 4.2: Reidemeister move 4 on a \widetilde{KTGD}	36
Figure 4.3: Connect sum	36
Figure 4.4: Unzip from two different views in \mathbb{R}^3	37
Figure 4.5: Two more generators needed for nonplanar KTGDs	38
Figure 4.6: The relation corresponding to the Reidemeister 1 move	39
Figure 4.7: The relation corresponding to the Reidemeister 2 move	40
Figure 4.8: The relation corresponding to the Reidemeister 3 move	41
Figure 4.9: The relation corresponding to the Reidemeister 4 move	42
Figure 4.10: The relation corresponding to the Reidemeister 5 move	43
Figure A.1: The pentagon relation, with the formulation given in Thurston [8] above and a version which highlights its similarity to the tetrahedron relation below	44
Figure A.2: The composition of relations for one half of the pentagon relation	45

ACKNOWLEDGEMENTS

Thanks to Dror Bar-Natan, Scott Morrison, Emily Peters, Justin Roberts, Peter Teichner, Dylan Thurston, Johnny and Amanda.

VITA

1999	B. S. in Mathematics, Geology <i>with honors</i> , Williams College, Williamstown
2001-2004	Graduate Teaching Assistant, University of California, San Diego
2004	M. A in Mathematics, University of California, San Diego
2004-2007	Graduate Student Instructor, University of California, Berkeley
2007-2008	Instructor, University of California, Berkeley
2008	Ph. D. in Mathematics, University of California, San Diego

PUBLICATIONS

Cox, R., Martin, M.W., Comstock, J.C., Dickerson, L.S., Ekstrom, I.L and Sammons, J.H. , “Sedimentology, stratigraphy and geochronology of the Proterozoic Mazatzal Group, central Arizona”, *Geological Society of America Bulletin*, 114, 2002.

Cox, R., Atkinson, R.K., Bear, B.R., Brandriss, M.E., Chokel, C.B., Comstock, J.C., Gutmann, E.D., Interest, L.B., Schildgen, T.F., Teplitzky, S.J., Willis, M.P., “Changes in a fringing reef complex over a thirty-year period: coral loss and lagoon infilling at Mary Creek, St. John, U.S. Virgin Islands”, *Bulletin of Marine Science*, 66, 2000

ABSTRACT OF THE DISSERTATION

A Finite Presentation of Knotted Trivalent Graphs

by

Jana Comstock

Doctor of Philosophy in Mathematics

University of California San Diego, 2008

Professor Justin Roberts, Chair

Professor Peter Teichner, Co-Chair

While knot theory has been studied since the 19th century (and arguably for thousands of years prior to Gauss), knotted trivalent graphs are objects of relatively recent interest. We will extend the methods used by Thurston to find generators of KTGs in 2002, and use them to determine a finitely generated list of relations, thereby acquiring a finite presentation of knotted trivalent graphs.

We will first define a set of knotted trivalent graph diagrams, and reiterate Thurston's result that they are generated by the diagrams for the tetrahedron and the twisted tetrahedron. We will then use a version of the same algorithm to establish that the relations on knotted trivalent graph diagrams are finitely generated by the four relations corresponding to the operations: disjoint union with a tetrahedron, disjoint union with a twisted tetrahedron, connect sum, and unzip.

Finally we will extend these results to knotted trivalent graphs themselves by equipping KTGs with a thickening of each edge and considering an extended list of Reidemeister moves for knotted trivalent graphs. By adding generators to account for twisting of thickened edges and relations corresponding to each of these five Reidemeister moves we will complete the finite presentation.

Chapter 1

Introduction

With the advent of the Kontsevich integral and its vast potential in the field of knot and link theory, knot theorists have begun to widen the scope of their research to include not only knots and the bordisms between them in their calculations, but also knotted trivalent graphs and the singular surfaces known as webs or foams which act as bordisms between knotted trivalent graphs. Reshetikhin and Turaev [7], Watanabe [10], and Bar-Natan et al [3] have defined manifold invariants with values in the space of these graphs. Further, if it can be shown that the relations on knotted trivalent graphs are contained in the quotient of the Kontsevich integral, it might significantly simplify the task of calculating these integrals. To this end the work presented here will not only provide a list of generators for knotted trivalent graphs, which has previously been accomplished in [8], but also a finite list of generators of relations.

In this paper we complete the details of finitely presenting knotted trivalent graphs, or KTGs. Previous work by Bar-Natan and Thurston [8] has determined generators for the algebra of KTGs and an algorithm for producing any KTG from these generators; we will adapt the algorithm slightly so as to more easily classify the relations on KTGs. KTGs will be described in greater detail in Chapter 4; for now, we will simply note that they are trivalent graphs which have been embedded into \mathbb{R}^3 (and thus knotted).

In Chapter 3 we will present a finite presentation for knotted trivalent graph diagrams rather than knotted trivalent graphs themselves. Our first step will be

to define knotted trivalent graph diagrams and a set of knotted trivalent diagram blueprints, and a host of other miscellany. The basic operations disjoint union, connect sum, and unzip will be defined not only for KTGDs but for blueprint versions as well. A number of convenient if not always critical operations will also be explained. The maximal tree which is the defining characteristic of these blueprints will act as a guide for applying the slightly altered version of Thurston's generating algorithm; this algorithm will yield the following theorem:

Theorem 1. *(Thurston, Bar-Natan): **KTGD** is generated by the tetrahedron and the twisted tetrahedron under the operations disjoint union, connect sum, and unzip.*

Next we will establish relations corresponding to each of our basic operations. When disjoint union is considered as a unary operation, there will be two versions of it on our basic list: disjoint union with a tetrahedron, and disjoint union with a twisted tetrahedron. This is equivalent to stating that **KTGD** is generated by those two diagrams. Along with the relations for connect sum and unzip, we will have the building blocks necessary to construct all possible relations on **KTGD**, yielding the next theorem:

Theorem 2. *Any relation between two sequences of operations is a composition of the four basic relations corresponding to disjoint union with a tetrahedron, disjoint union with a twisted tetrahedron, connect sum, and unzip.*

Together these results give a finite presentation for knotted trivalent graphs diagrams, but not for knotted trivalent graphs themselves. In Chapter 4 we will equip knotted trivalent graphs with thickened edges which contain twisting information; this allows the basic operations to be extended to knotted trivalent graphs as well as diagrams. These thickenings also make for a larger possible set of knotted trivalent graphs diagrams, and two more generators (or two more versions of disjoint union, if you prefer) must be added to our list of generators (and operations).

Additionally, the relations for knotted trivalent graph diagrams fail to account for those sequences of operations which yield two distinct knotted trivalent graph diagrams nonetheless corresponding to the same knotted trivalent graph. These

diagrams differ by a sequence of extended Reidemeister moves [4, 11], so adding a relation for each of the five moves in the extended Reidemeister list for knotted trivalent graphs give all possible relations on knotted trivalent graphs.

Theorem 3. *(Kauffman): Ambient isotopy of knotted trivalent graphs is generated by the moves shown in section 2.1. That is, any two diagrams of isotopic knotted trivalent graphs are related by a finite sequence of these moves.*

Corollary. *All Reidemeister moves on **KTGD** map to the identity on **KTG**, and the relations thus generated are the only ones required to extend our list of relations to **KTG**.*

Combining these statements, we have the main result: that knotted trivalent graphs are finitely presented.

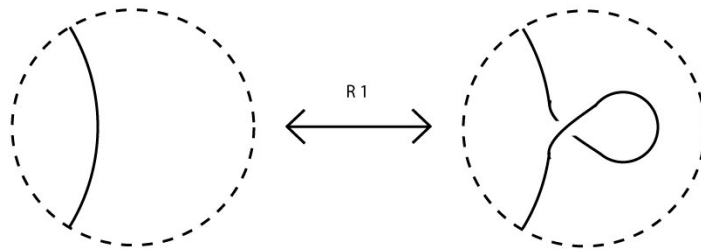
Chapter 2

Background

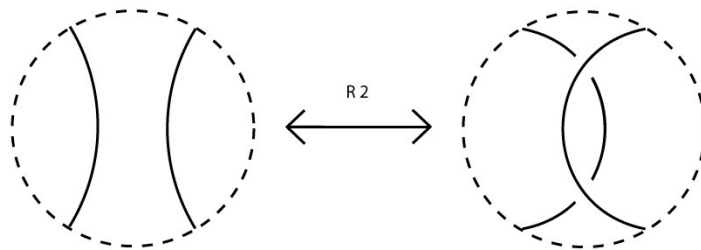
2.1 Reidemeister moves

In 1926, Reidemeister (as well as Alexander and Briggs) [6, 1] proved that the familiar list of three moves generates isotopy of knots using an essentially combinatorial method. Kauffman [4] has extended this list to a set of five Reidemeister moves which are sufficient to generate isotopy between graphs. In addition to the usual three moves:

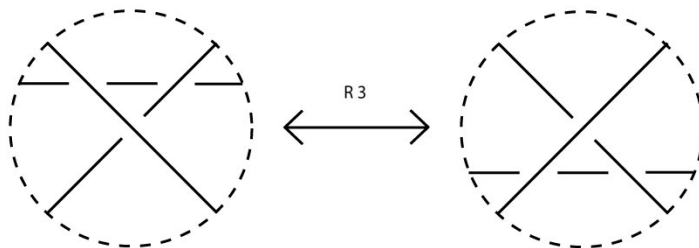
- Reidemeister 1



- Reidemeister 2

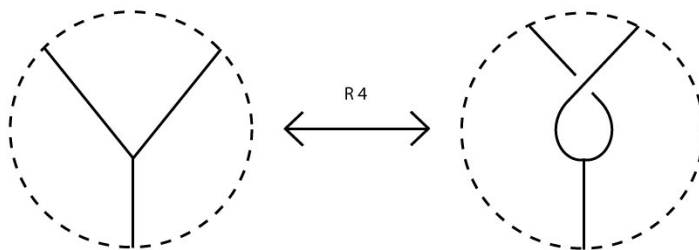


- Reidemeister 3

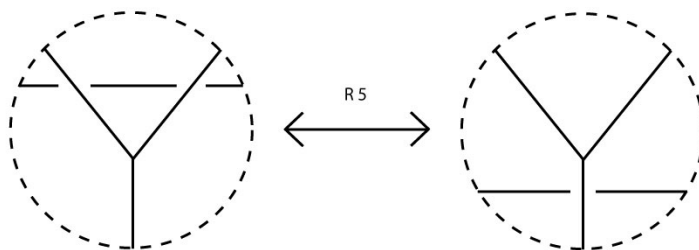


Kauffman [4] includes another two moves for graphs, pulling a strand under (or over) a vertex, and twisting a vertex:

- Reidemeister 4



- Reidemeister 5



The moves shown here are for trivalent graphs; Kauffman [4] and Yamada [11] give lists for more general sets of graphs.

2.2 The Kontsevich Integral

The Kontsevich integral, whose merit as the “universal finite type invariant” is the ability to yield any finite type knot invariant by applying some weight system, has been thoroughly discussed in many fine papers [5, 2]; rather than attempting to reproduce their achievements we will relate a slightly different interpretation

pioneered by Dylan Thurston in his undergraduate thesis and expounded to me via Dror Bar-Natan and Scott Morrison. This “Stonehenge” idea (so called because of the “stellar coincidences” counted therein) is that the Gauss integral can be generalized to the Kontsevich integral.

The Gauss integral calculates the linking number, which can be thought of as counting configurations of certain additions to the embedded link; the (oriented) number of vertical intervals one can draw connecting one component of a link to the other.

Thurston’s Stonehenge formula gives a linear combination of chinese characters. A chinese character is a univalent graph with its univalent vertices lying on a distinguished circle. While we remember a cyclic order at each vertex, and we remember the orientation of the distinguished circle, this chinese character is not considered to be embedded in the plane, and any tetravalent vertices are simply introduced by the necessity of drawing the diagram in the plane. The coefficients of the chinese character diagrams turn out to be exactly the number of configurations of each if the direction of each segment in the diagram is determined by placing a star on the sphere at infinity, at which the segment must point.

2.3 TG Algebras

A TG algebra has as its spaces trivalent graphs and as its morphisms the operation connect sum, which acts on two arcs of disjoint components, and the operation unzip, which acts on an arc. Delete is often listed as an operation as well, although it can be obtained via a sequence of connect sums and unzips, as will be shown in Chapter 1. If we allow the set of objects associated with each trivalent graph to be its embeddings into \mathbb{R}^3 , knotted trivalent graphs are an obvious example of a TG algebra, and perhaps the most natural/motivating/primeval one. The finite presentation of KTGs now yields a short list of relations which one must check are in the quotient of the Kontsevich integral in order to extend the Kontsevich integral to a TG morphism. This motivates the result presented in this paper.

2.4 The Århus Integral and KTG manifold invariants

Just as the Kontsevich integral is the universal finite type invariant for knots, the Århus integral is the universal finite type invariant for manifolds[3]. It computes the Le-Murakami-Ohtsuki invariant, the manifold theoretical version of the Vassiliev invariant, which takes values in the space of trivalent graphs. As the analogue of the Kontsevich integral, the information computed by the Århus integral encompasses that which is provided by 3-manifold invariants such as the Turaev-Viro invariant [9].

Chapter 3

Knotted trivalent graph diagrams

Most of the work of determining generators and relations will be executed in this chapter, but with respect to knotted trivalent graph diagrams rather than knotted trivalent graphs themselves. In Chapter 3 these results will be translated to knotted trivalent graphs. In this chapter we will use an adaptation of the algorithm employed by Thurston [8] to prove that knotted trivalent graphs are finitely generated, where the slight adaptations provide a basis for proving that the relations on knotted trivalent graph diagrams are also finitely generated, with generators of relations corresponding to the generators of knotted trivalent graphs themselves.

3.1 Sets

First we must define the several slightly different types of objects under consideration, including knotted trivalent graphs, knotted trivalent graph diagrams, diagrams with a preferred edge, blueprint diagrams, and blueprints with a preferred edge.

Definition 1. A knotted trivalent graph or KTG is a set of vertices connected by edges such that there are three edges at each vertex, along with an embedding into \mathbb{R}^3 . For our purposes we will not consider these objects to be oriented; we will eventually equip them with a framing but will initially ignore framing. We will

often refer to the set of all such possible objects, up to isotopy, as KTGs or **KTG**.

Notice that some of the usual requirements for graphs do not apply; loops and multiple edges are allowed. It is also worth remarking that this definition is slightly awkward, as examples cannot be illustrated without first being projected into \mathbb{R}^2 . Therefore we will begin by working with diagrams of KTGs and will extend our results to KTGs in Chapter 3.

Definition 2. A knotted trivalent graph diagram or KTGD is a projection of a KTG into S^2 up to isotopy; as with diagrams of knots, this projection consists of an immersion where the over/under crossing information is retained at the double points by drawing the strand which passes under as broken. The occurrence of these double points renders the KTGD a 3-, 4-valent graph, and edges are considered to be the arcs between vertices and/or crossings. The edges of this KTGD are classified as either inner or outer edges, where outer edges are those bounding the exterior component of the complement. These objects are referred to collectively as KTGDs or **KTGD**.

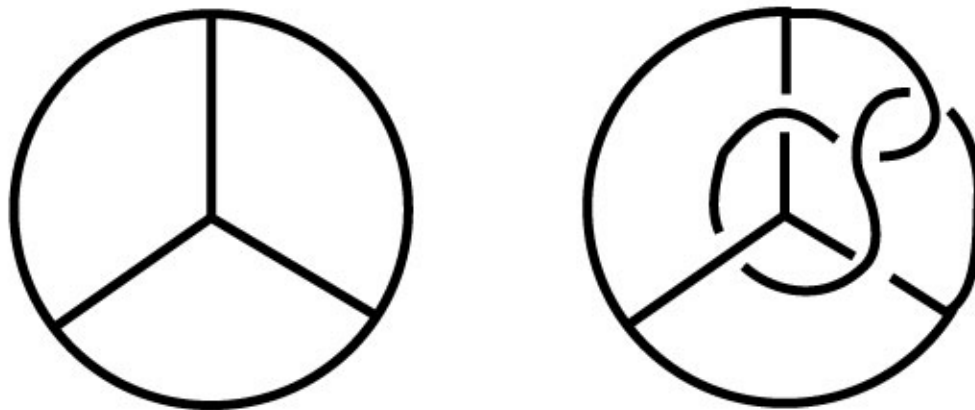


Figure 3.1: Two KTGDs which represent different KTGs in 3-space

In order to define the operations on knotted trivalent graph diagrams carefully, it will be necessary to refer to yet another set of objects:

Definition 3. The set of KTGDs with a preferred edge selected will be referred to as **KTGE**.

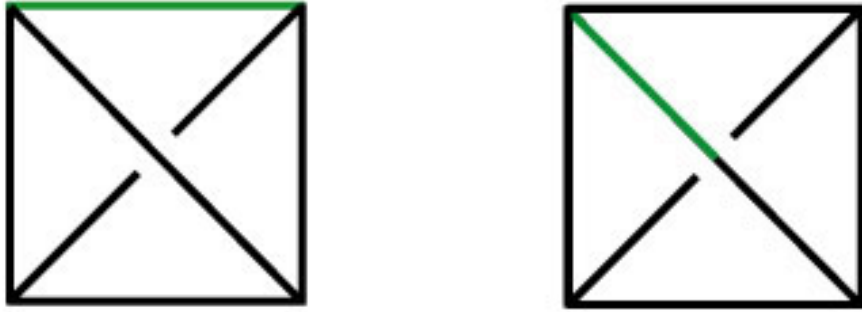


Figure 3.2: Two KTGEs which represent the same KTGD with preferred edge indicated in green

After defining operations and finding a list of generators for \mathbf{KTGD} , we will still need to give the set of the blueprints used in the generating algorithm a name so we can consider the relations on \mathbf{KTGD} :

Definition 4. A knotted trivalent diagram blueprint or $\widetilde{\mathbf{KTGD}}$ is a KTGD along with the choice of a maximal tree on the underlying 3-, 4-valent knotted trivalent graph diagram. We will call the set of such objects $\widetilde{\mathbf{KTGD}}$.

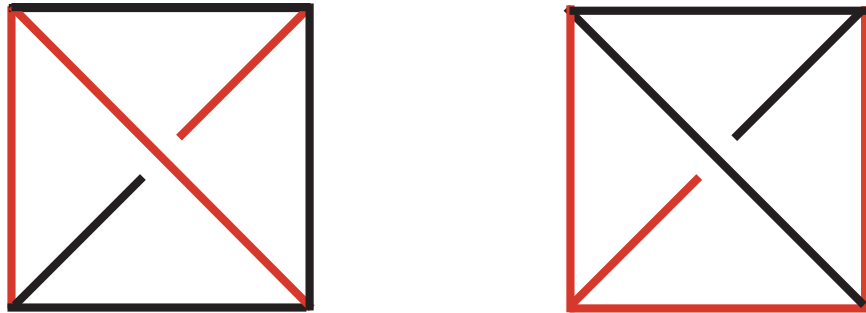


Figure 3.3: Two $\widetilde{\mathbf{KTGD}}$ s which represent the same KTGD with maximal tree indicated in red

Finally, we have an analogue for KTGDs with a preferred edge as well:

Definition 5. The set of $\widetilde{\text{KTGD}}$ s with a preferred edge selected will be referred to as $\widetilde{\text{KTGE}}$.

The relationships between these sets is illustrated in the diagram below.

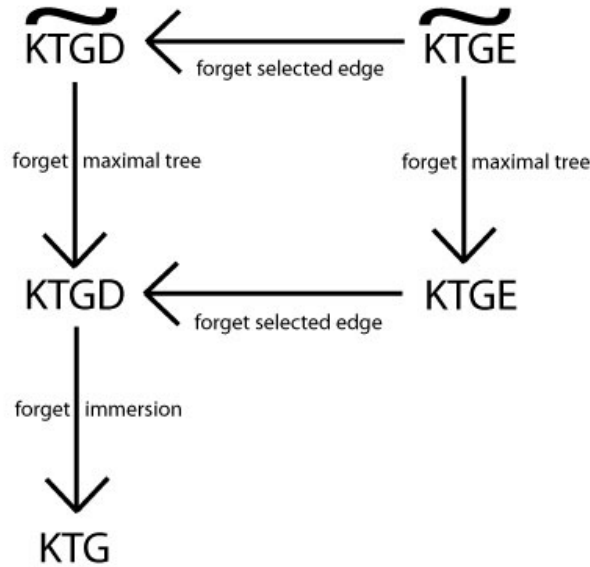


Figure 3.4: Maps between sets

3.2 Operations

Now we can introduce the operations on KTGD s. There are three basic operations:

- Connect sum is a binary operation $\# : \text{KTGE} \times \text{KTGE} \rightarrow \text{KTGD}$ where the selected KTGE s must be separate components and the selected edges must be exterior edges. Both arcs are cut and the ends are reattached to the ends from the other arc as indicated by the planar embedding. This operation is illustrated in Figure 3, where any portion of the KTGD s which lies outside the dotted circle is unchanged. Note that the input arcs must be from two disjoint KTGD s, or else the choice of how to splice them together is not well defined.

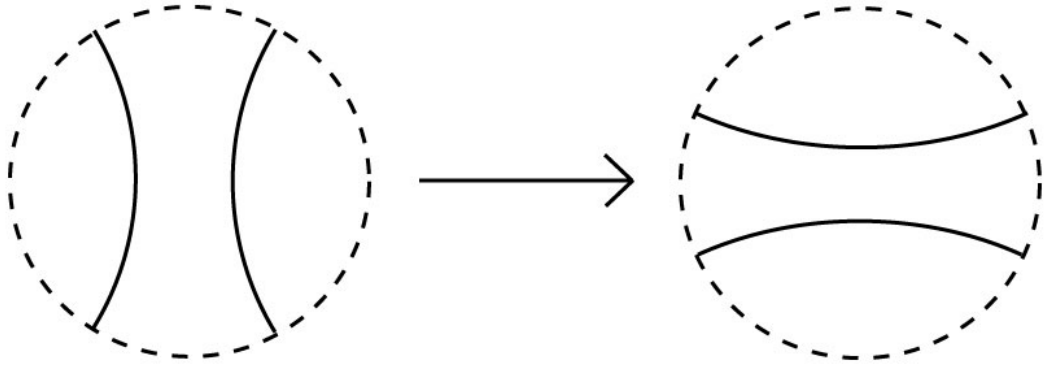


Figure 3.5: Connect sum

- Unzip is a unary operation $u: \mathbf{KTGE} \rightarrow \mathbf{KTGD}$ which splits an edge connecting two distinct vertices. Unzip removes an entire edge and its endpoints by connecting the other edges at those vertices to one another, again as indicated by the planar embedding. Unzip is illustrated in figure 4, with areas outside the dotted circles remaining unchanged.

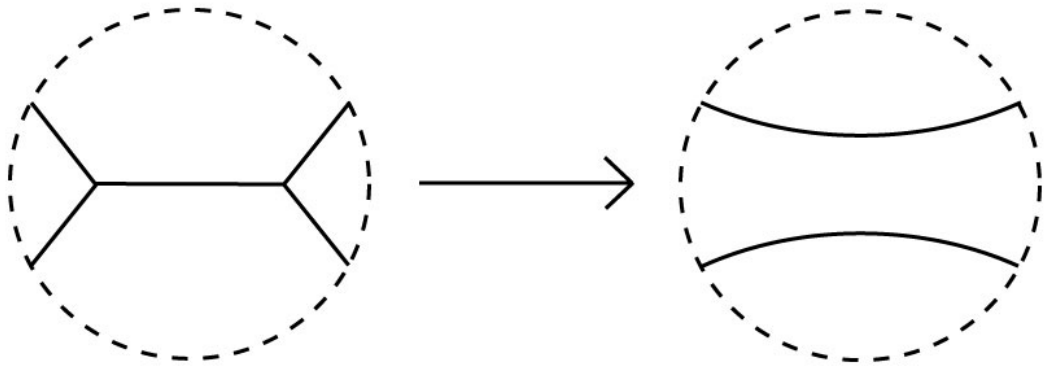


Figure 3.6: Unzip

- Disjoint union is a binary operation $\sqcup : \mathbf{KTGD} \times \mathbf{KTGD} \longrightarrow \mathbf{KTGD}$ which simply places two KTGDs adjacent to one another in the plane.

There is also a variant of connect sum used by Bar-Natan; one advantage of this formulation is that it makes apparent the necessity of choosing disjoint components in order for connect sum to be well defined:

- An alternate connect sum adds one vertex to each chosen edge and creates a new edge to connect them; this move can be accomplished with our original definition of connect sum by connect summing a pair of spectacles between the two KTGDs:

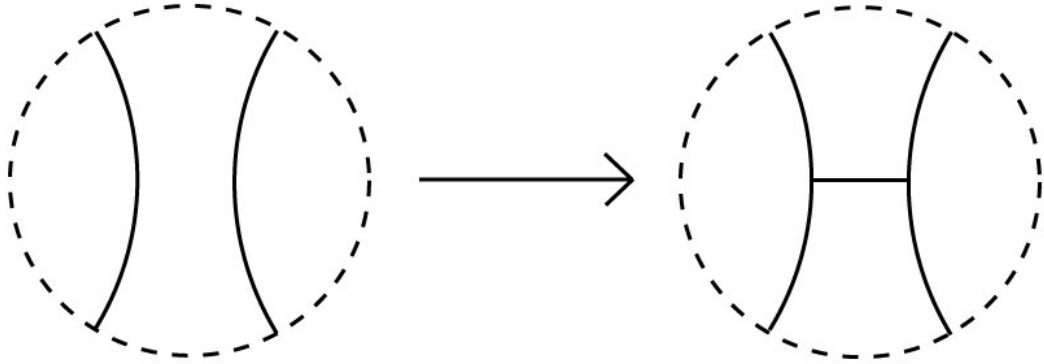


Figure 3.7: Alternate connect sum

Our operations on $\widetilde{\mathbf{KTGD}}$ will include three switch tree operations as well as the familiar KTGD operations:

- The “switch leaf” operations for 3- and 4-valent vertices changes which edge attached to a particular vertex is included in the maximal tree. Since all vertices outside the dotted circle must be included in this maximal tree, any choice of an edge coming into this vertex connects it to the tree and creates a valid maximal tree.

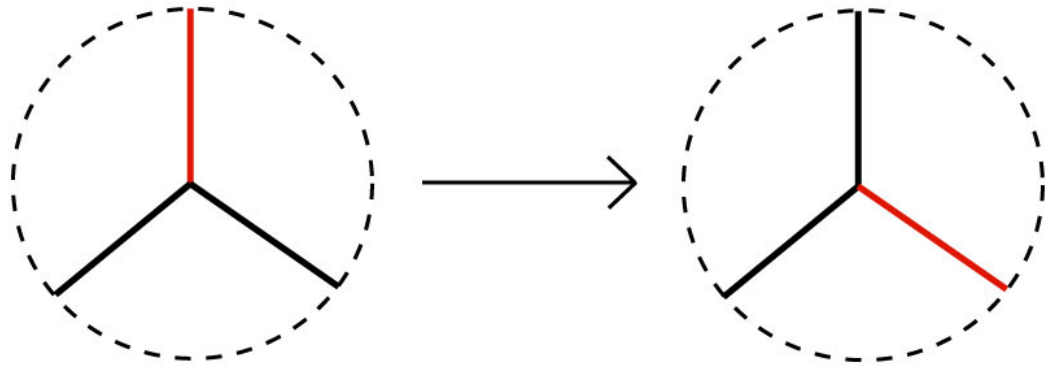


Figure 3.8: Switch leaf for a trivalent vertex

Notice this can only be applied to vertices which do not connect to both ends of the same edge, or there would necessarily be two new edges replacing the old one. This will never be a problem, as the switch leaf operation will only be used to move from one choice of maximal tree to another.

- The “switch branch” operation changes which edge connecting two subtrees of the maximal tree is used to connect them:

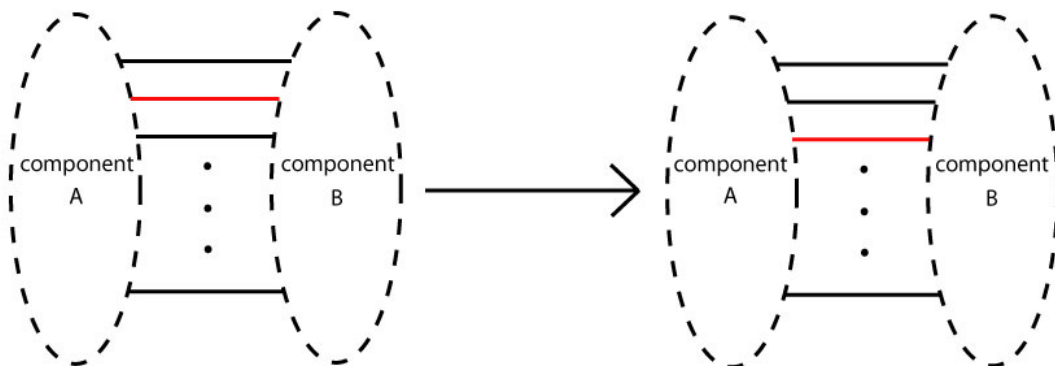


Figure 3.9: Switch branch

In this case the area inside the dotted circles remains unchanged. The placement of the circles is altered to emphasize the fact that since these are possible branches

of a maximal tree they connect two otherwise disjoint components, labeled here as component A and component B.

These two “switching” operations add an interesting feature to our diagram of KTG relations:

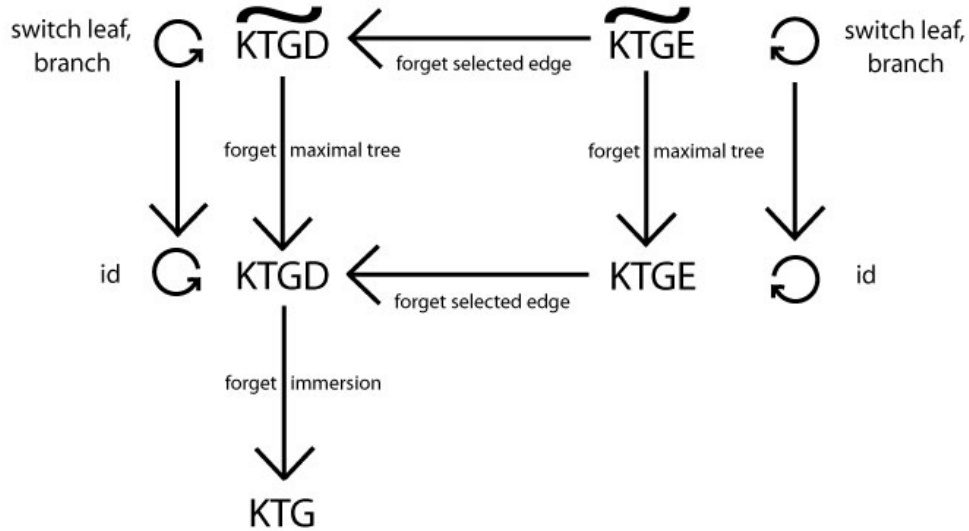


Figure 3.10: Maps between sets

Since we are forgetting the maximal tree when we go from $\widetilde{\mathbf{KTGD}}$ or $\widetilde{\mathbf{KTGE}}$ to \mathbf{KTGD} or \mathbf{KTGE} , the switch leaf/branch operations map to the identity.

- The operation disjoint sum twiddle $\sqcup : \widetilde{\mathbf{KTGD}} \times \widetilde{\mathbf{KTGD}} \rightarrow \widetilde{\mathbf{KTGD}}$ is identical to \sqcup except that it acts on objects in $\widetilde{\mathbf{KTGD}}$ instead of \mathbf{KTGD} .
- The operation connect sum twiddle $\# : \widetilde{\mathbf{KTGE}} \times \widetilde{\mathbf{KTGE}} \rightarrow \widetilde{\mathbf{KTGD}}$ acts on two $\widetilde{\mathbf{KTGE}}$ s where the preferred edges are both exterior, from disjoint components, and not part of the maximal tree; one of the two new edges created is added to the maximal tree of the $\widetilde{\mathbf{KTGD}}$ thus created. Alternatively connectsum may act on two edges from separate components exactly one of which is included in a maximal tree; in this case both output strands are included in the maximal tree for the output.

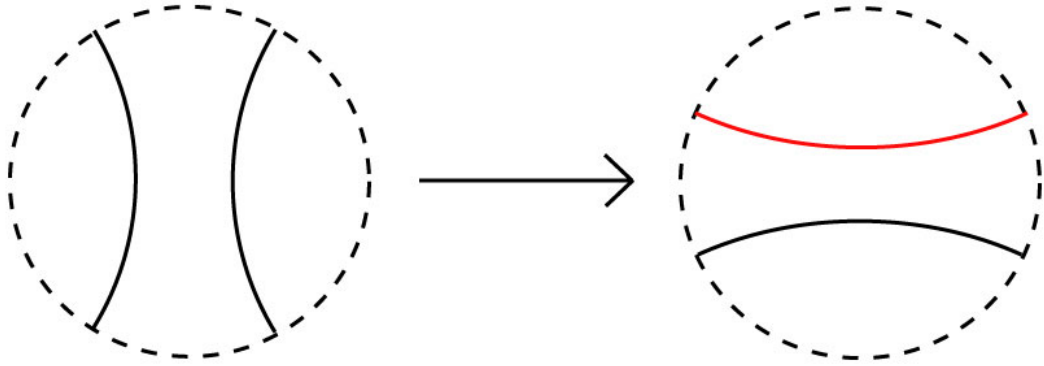
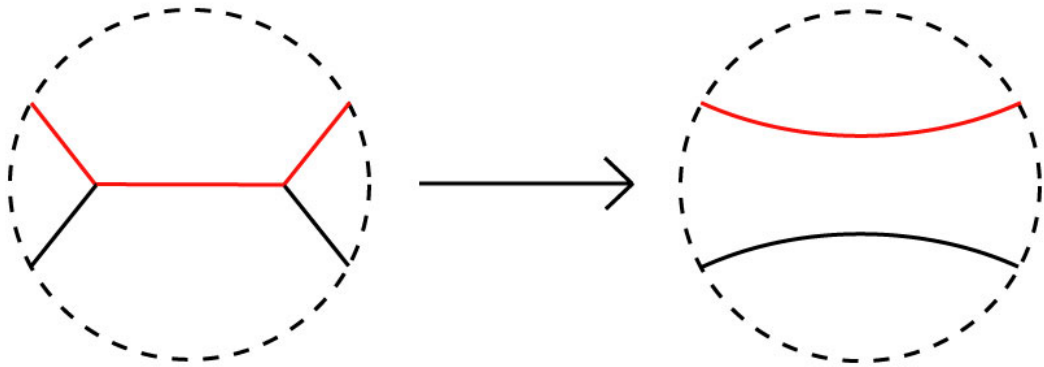


Figure 3.11: Connectsum

- The operation $\text{unzip } \tilde{u} : \widetilde{\mathbf{KTGE}} \longrightarrow \widetilde{\mathbf{KTGD}}$ acts on a $\widetilde{\mathbf{KTGE}}$ where the selected edge is not only in the complement of the maximal tree, but the trivalent vertex at each end of the selected edge has only one edge in the maximal tree; in this case neither of the output strands are included in the resulting maximal tree. Alternatively, the selected edge may be part of the maximal tree and the upper two(or lower two) attached edges may be on the maximal tree; in this case the upper (or lower) output strand is included in the resulting maximal tree:

Figure 3.12: An example of $\widetilde{\text{unzip}}$

As we may change the chosen maximal tree at will using the switch leaf and switch branch operations, it is always possible to rearrange the tree so the selected edge

satisfies these criteria as long as the unzip doesn't split the $\widetilde{\text{KTGE}}$ into two disjoint KTGD components with vertices in each. Fortunately such a split is obtainable through other means. The simplest explanation for the defensibility of excluding this particular type of unzip is that any KTGD can be generated without it, as will be shown.

There are several additional operations on **KTGD** which, while extraneous in the sense that each may be obtained as a composition of the operations listed above, are often convenient. If disjoint union is considered as many unary operations, with one for each possible KTGD, rather than a single binary operation then it will be shown that the only two which properly belong in the list of basic operations above are disjoint union with a tetrahedron and disjoint union with a twisted tetrahedron, and all the other disjoint unions fall into the additional category, to which we will now add a few more operations:

- Kill removes a component with no vertices (an unknot). Notice that if there are any other components this is equivalent to connect summing with them.

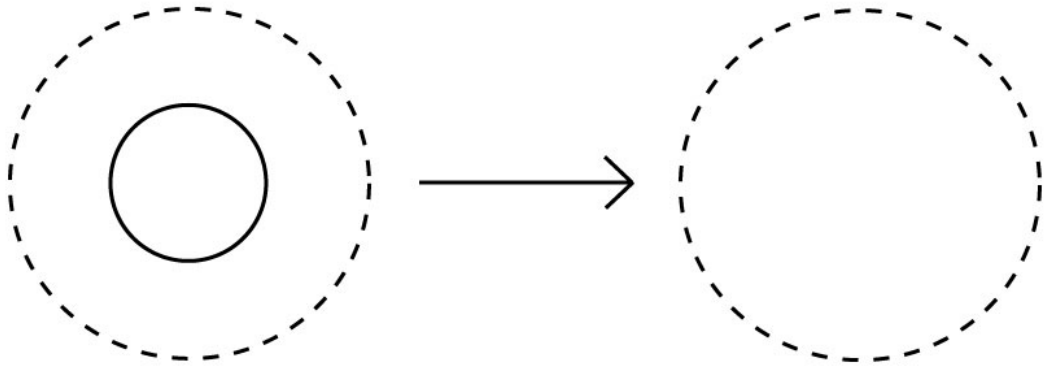


Figure 3.13: Kill

- Bubble, which appears on Thurston's list [8] of operations along with the observation that this operation can be accomplished by connect summing with a theta, splits an edge apart in the middle, adding two vertices:

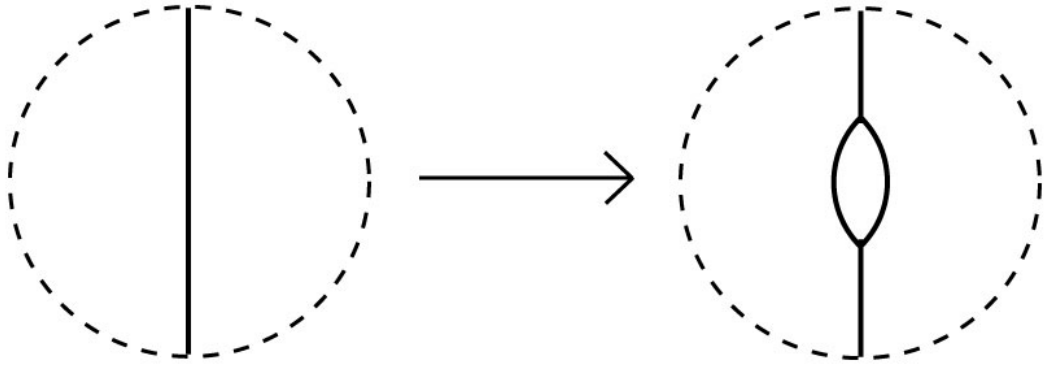


Figure 3.14: Bubble

- Vertex connect sum is a binary map from the space of KTGDs with a selected vertex to **KTGD**, again acting on vertices from two disjoint components. It is equivalent to a connect sum either between the two edges above the vertex or the two below it, followed by an unzip on the resulting edge:

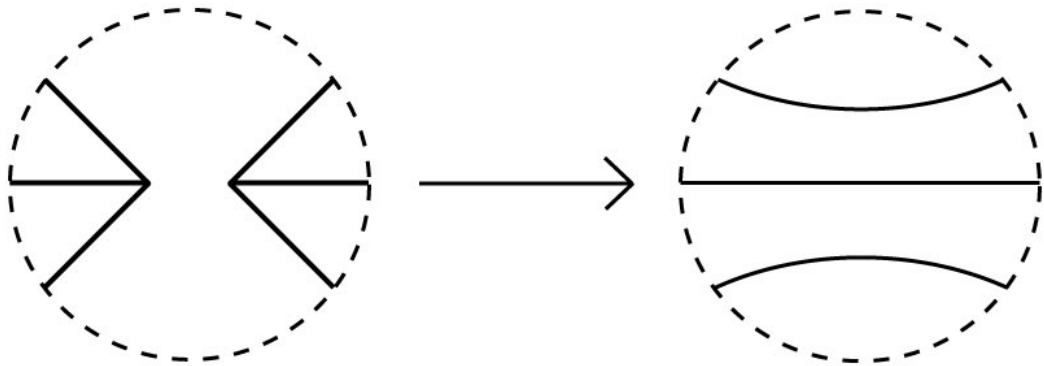


Figure 3.15: Vertex connect sum

Vertex connect sum will be used in the generating algorithm for KTGDs. Extending this composition of a connect sum and subsequent unzips to a larger subgraph than just one vertex, we have:

- Tree connect sum is a binary map from the space of KTGDs with a selected tree (both selected trees must be identical) to **KTGD**, where the chosen

trees must be from disjoint components. It is equivalent to a connect sum on any edge of the first tree and the corresponding edge from the second tree, followed by all possible unzips.

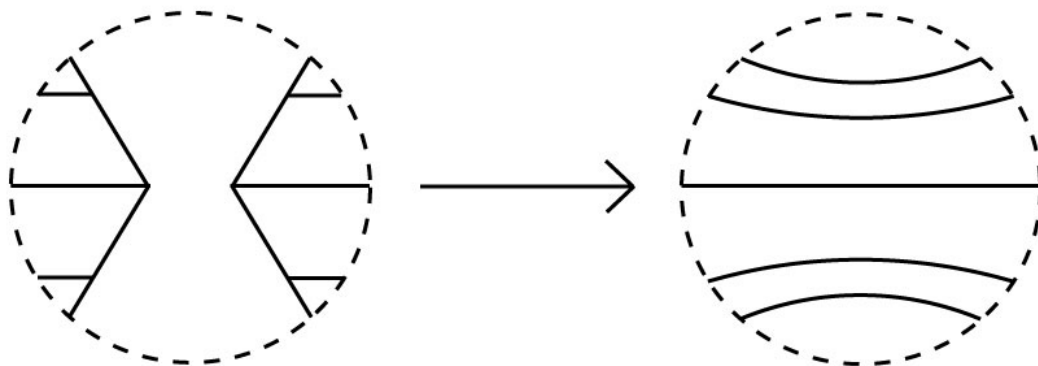


Figure 3.16: Tree connect sum

There are inverses for disjoint union, connect sum, and tree connect sum (which is to say a connect sum followed by unzips may still be inverted), but only a partial inverse for unzip. Disjoint union, when considered as a unary operation, is invertible simply by performing as many unzips as are possible and killing the resulting circles:

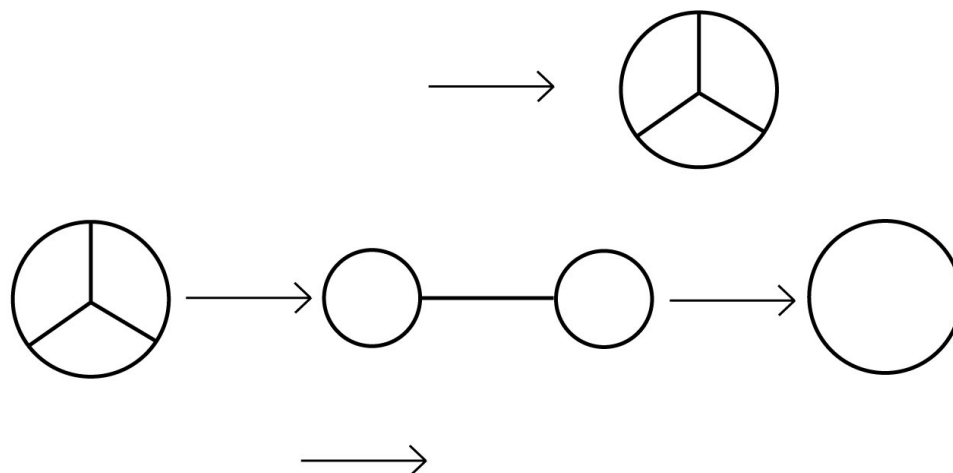


Figure 3.17: The disjoint union in the top line is undone by the sequence of unzips and kills below.

Connect summing a second KTG onto any KTG is also invertible simply by unzipping all edges which were contributed by the second KTG and then performing a disjoint union with that KTG, and a tree connect sum is invertible by connect summing with the mirror image of the second KTG, performing as many unzips on the two added KTGs as possible (in other words performing a tree connect sum with as large a chosen tree as possible), and then reconstructing the second KTG as a disjoint component. An example is given in Figure 2.18.

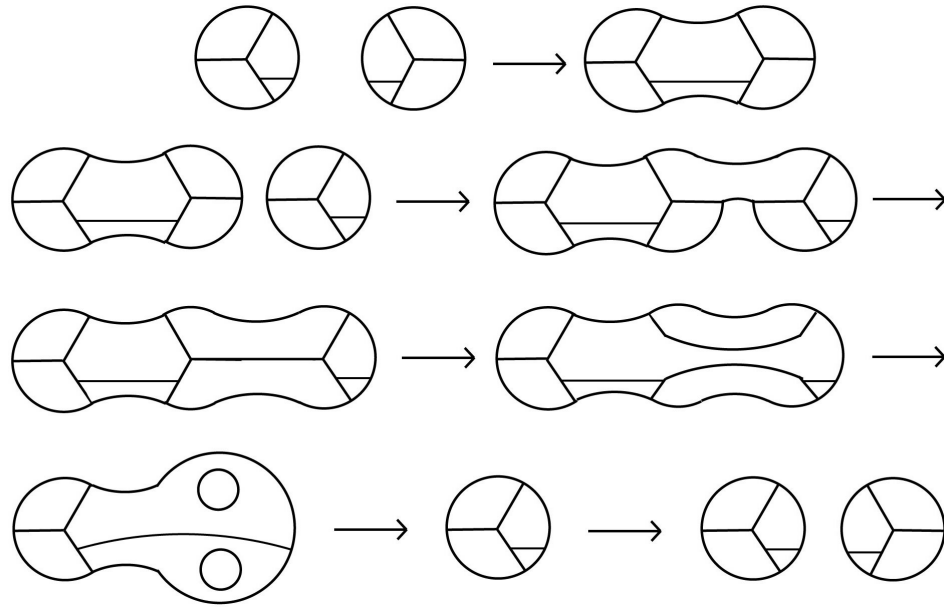


Figure 3.18: The connect sum in the top line is undone by the connect sum, sequence of unzips, and disjoint union below.

Unzip is only invertible when the unzip splits the KTGD into two separate components; in this case one can reconnect them by connect summing a theta between them, as shown in Figure 2.19. This procedure cannot be followed if the unzip does not split the KTGD into two components, as the second connect sum would not be performed on distinct components and would therefore not be well defined.

The invertibility of unzips which split a KTGD into two separate components, unlike other unzips, allows us, if we prefer, to consider them as not properly unzips at all, as we do in the $\widetilde{\text{unzip}}$ case. The invertibility reflects the fact that there is an alternate method of effecting this result and that we need not include the possibility

of a "splitting" unzip.

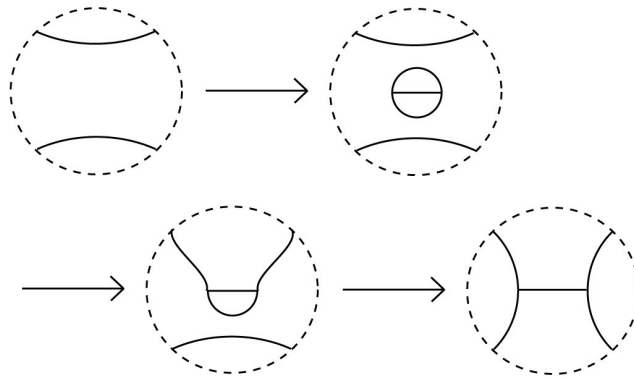


Figure 3.19: Inverting an appropriate unzip

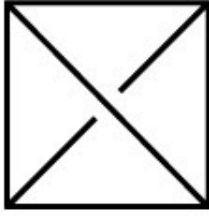
3.3 Generators

We now have the necessary machinery to devise an algorithm for constructing any KTGD by applying connect sum and unzip to the tetrahedra and twisted tetrahedra which, as we will show, generate KTGD. The algorithm we will present here is an adaptation of the algorithm described by Thurston in [8]. In addition to using the operations disjoint union, connect sum, and unzip described above, we will also refer to the following KTGDs, which the algorithm will show are generators for **KTGD**:

- tetrahedron



- twisted tetrahedron



Using these tetrahedra and twisted tetrahedra as generators, we may now construct any desired KTGD by applying vertex connect sums and unzips to tetrahedra and twisted tetrahedra as dictated by the following algorithm:

- Step 0: Select a $\widetilde{\text{KTGD}}$ which maps to the desired KTGD. In other words, recalling that a KTGD is actually a 3-, 4-valent graph with 4-valent vertices the crossings, choose any maximal tree on this 3-, 4-valent graph. We will often call this object a blueprint, since it is only a reference as to how to apply the following steps; it is not one of the objects we will operate on in order to generate the KTGD in question.

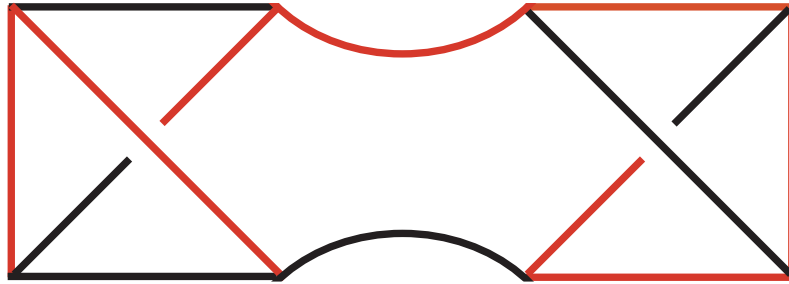


Figure 3.20: A possible blueprint (step 0) for



- Step 1: Beginning with the empty set and using the blueprint $\widetilde{\text{KTGD}}$ as a guide, perform a disjoint union with a tetrahedron for each 3-valent vertex

and a disjoint union with a twisted tetrahedron for each 4-valent vertex (crossing) in your blueprint.

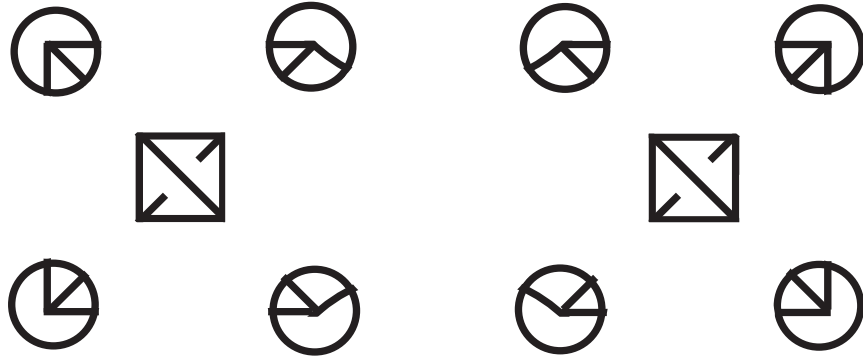
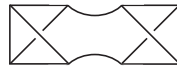


Figure 3.21: The completion of step 1 for



- Step 2: Perform a vertex connect sum along each edge in the maximal tree of the blueprint. Before each vertex connect sum on an edge the edge in question does not exist in the KTGD we are building, but after the vertex connect sum it does exist. Notice that we have performed the maximum possible number of connect sums, as there are no more disjoint components.

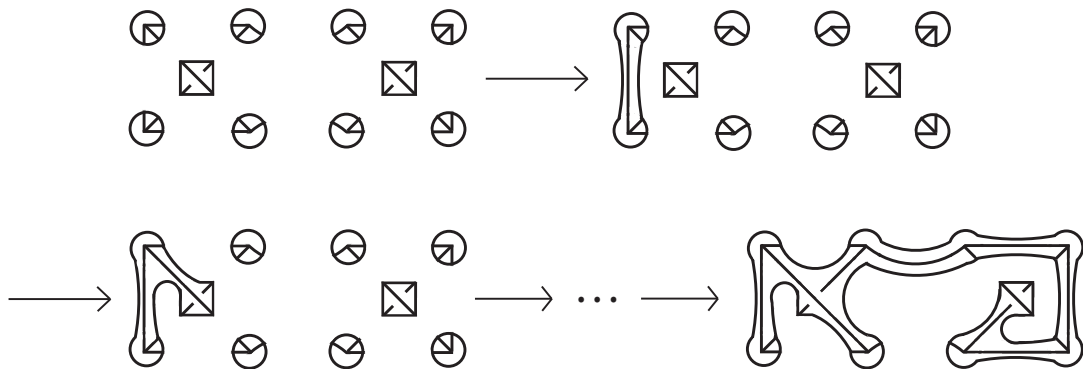


Figure 3.22: Step 2

- Step 3: Perform an unzip corresponding to each edge in the complement of the blueprint’s maximal tree. Again, these edges are created in the KTGD by each unzip.

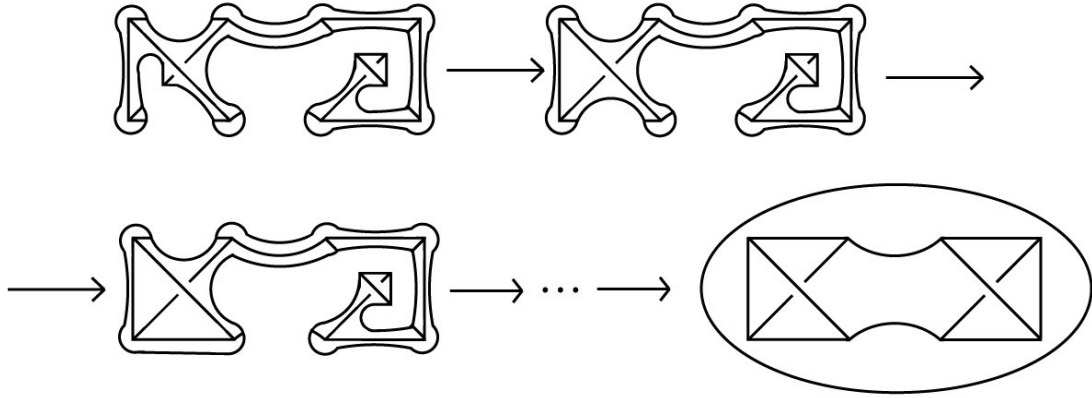


Figure 3.23: Step 3

- Step 4: Kill the extraneous boundary component.

Notice in particular that we never use a “splitting” unzip until the very last step when we split off the final boundary component, an unknot.

Every 3-valent vertex is created in step 1a, with some boundary component, and every 4-valent vertex is created in step 1b, with some boundary component. Every edge in the maximal tree is created in step 2, while the many boundary components are simultaneously reduced to one component (and the object becomes connected). In step 3, every other edge is created and the boundary component is separated from the desired KTGD, and step 4 removes the extraneous boundary component. We may now state with confidence the following theorem, published by Thurston [8] but jointly conceived by Thurston and Bar-Natan:

Theorem 1. (*Thurston, Bar-Natan*): **KTGD** is generated by the tetrahedron and the twisted tetrahedron under the operations disjoint union, connect sum, and unzip.

This follows directly from the algorithm.

Corollary. *Any two choices of maximal tree for a $\widetilde{\text{KTGD}}$ will yield the same KTGD when the algorithm is applied.*

3.4 Relations

We will now use this algorithmic procedure for generating a KTGD as a standard of comparison between all other possible methods of generating the same KTGD , giving us all possible relations. As a starting point, we define relations corresponding to the generators and operations used in our algorithm.

Lemma 1. *The relation between disjoint union with a tetrahedron and the standard algorithmic method of constructing a tetrahedron can be represented by the following commutative diagram:*

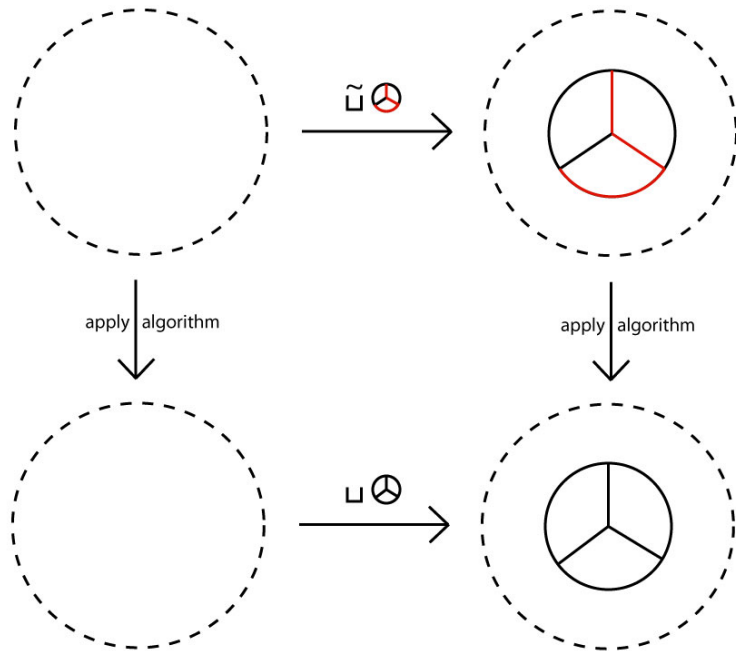


Figure 3.24: Tetrahedron relation

The objects in the top row of Figure 2.23 are $\widetilde{\text{KTGD}}$ s, and the objects in the bottom row are KTGD s. This is consistent with the notion that the algorithm starts with a blueprint and results in a KTGD . While the forgetful map from

$\widetilde{\text{KTGD}}$ to KTGD would yield the same output, the more complicated sequence of steps used in the algorithm is necessary here, and the disjoint union in the bottom row is only compared to steps 1 through 4 of the algorithm shown in the right hand column. In other words, if the area outside the dotted circles is the empty set, we are comparing two sequences of operations which begin with the empty set and end with the tetrahedron.

This diagram is actually only an example, as the tetrahedron relation encompasses not only this choice of maximal tree but any other choice as well, so there are fifteen different top rows we could draw corresponding to the fifteen choices of maximal tree. We will bear this in mind as we examine the next three relations, and afterwards introduce a commutative diagram for the switch tree operation to circumvent this complication.

Lemma 2. *The relation between disjoint union with a twisted tetrahedron and the algorithmic construction of the twisted tetrahedron can be represented by the following commutative diagram:*

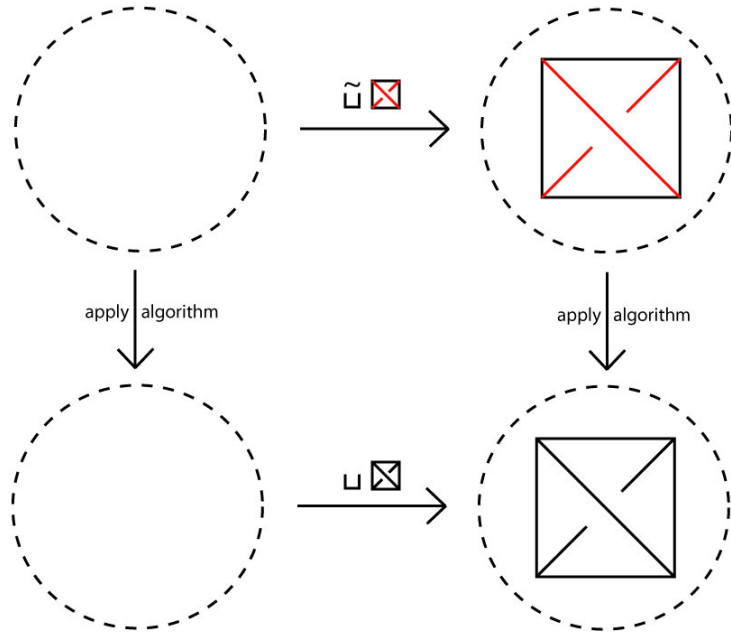


Figure 3.25: Twisted tetrahedron relation

Lemma 3. *Applying the algorithm to two blueprints and then connect summing them is equivalent to performing a $\widetilde{\text{connectsum}}$ on the same two blueprints and then applying the algorithm to the result; in other words, the following diagram is commutative:*

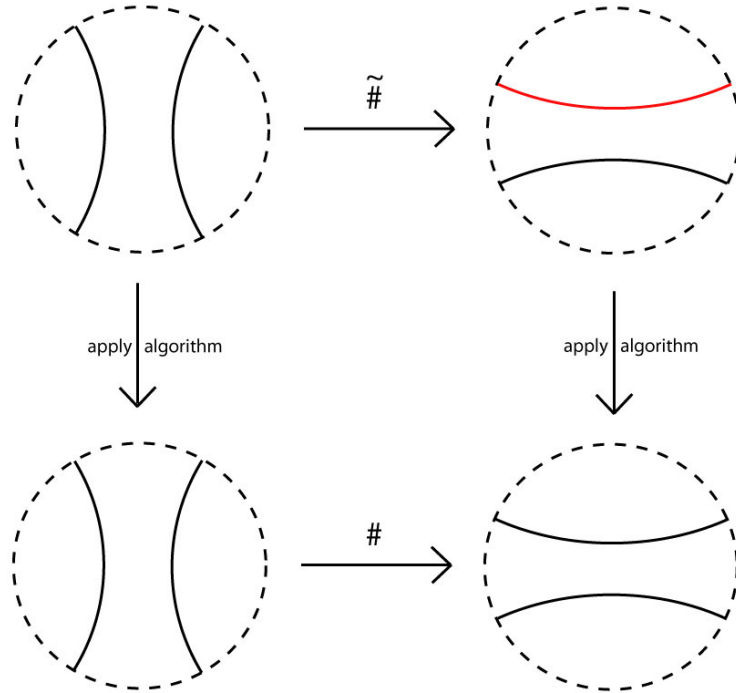


Figure 3.26: Connect sum relation

Lemma 4. *Applying the algorithm to a blueprint and then unzipping an edge is equivalent to performing a $\widetilde{\text{unzip}}$ on the same blueprint and then applying the algorithm to the result; in other words, the following diagram is commutative:*

As previously mentioned, each of the above four relations is actually only an example, and encompasses a number of different possibilities depending on the choice of maximal tree in the top row. If we are not comfortable considering the relation to generally represent any of these choices, we can distinguish one canonical choice as the relation and acquire all other versions by adding a commutative triangle for the switch leaf operation and one for the switch branch operation, allowing us to alter the maximal tree in the top row as necessary. The commutative triangle for the switch leaf operation is shown in Figure 2.27.

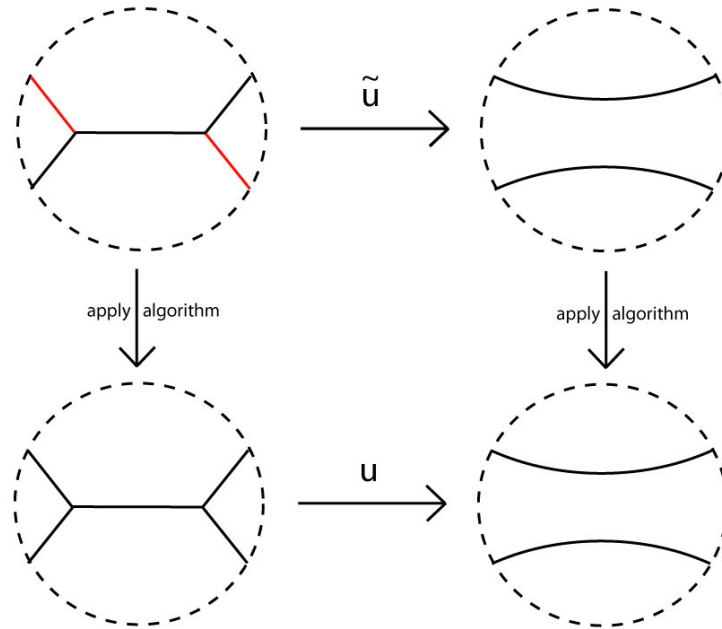


Figure 3.27: Unzip relation

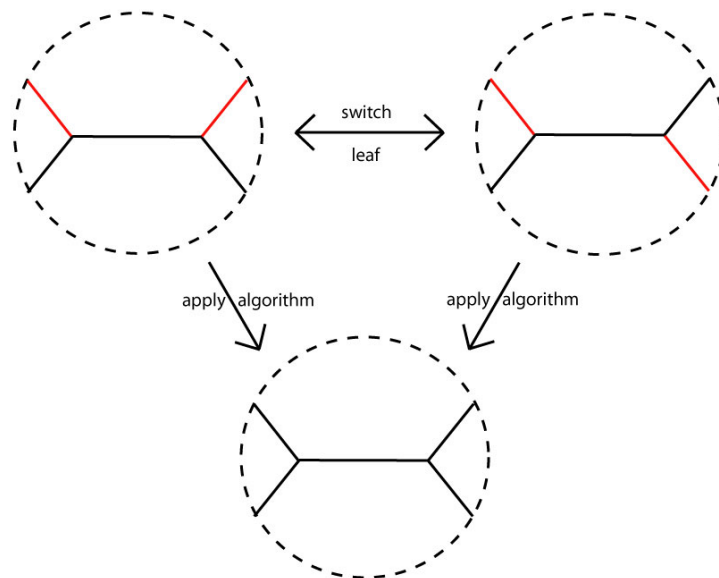


Figure 3.28: A commutative triangle reconciling different formulations of the unzip relation

Any composition of these four basic relations yields a new relation between the composition of operations represented in the bottom row and the algorithm which yields the same KTGD. An example is shown in Figure 2.29.

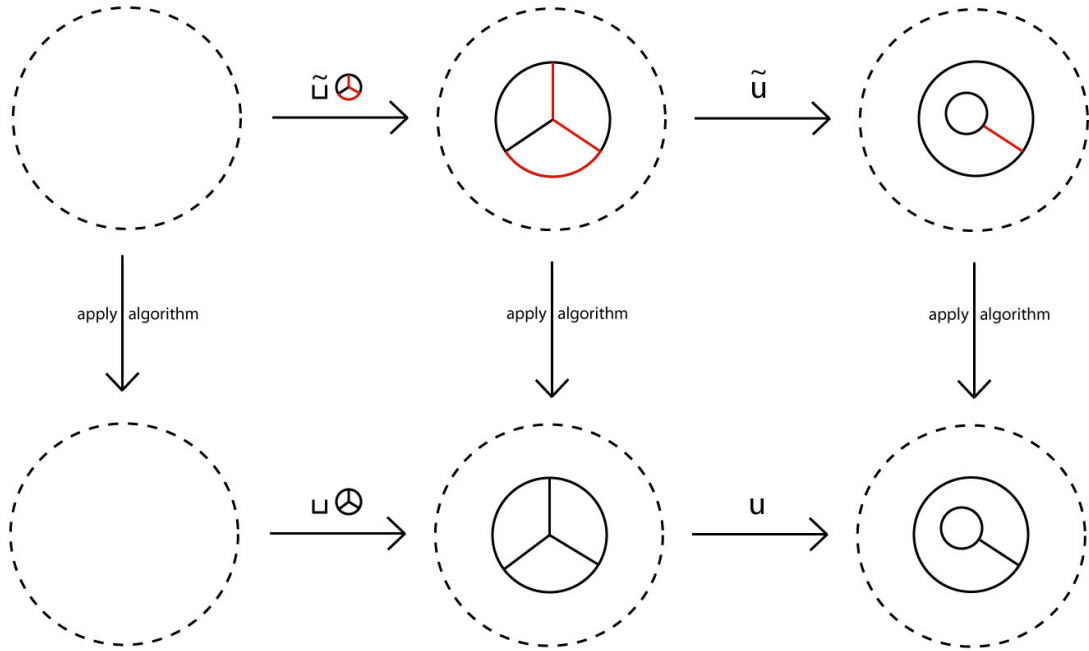


Figure 3.29: A composition of relations

Theorem 2. *Any relation between two sequences of operations is a composition of these four basic relations.*

We can show this by relating any sequence of operations which results in a particular KTGD to the algorithmic sequence which yields this KTGD. Given any sequence of operations, we simply compose the relations corresponding to these relations. The resulting chain of commuting squares is a relation between the sequence (which is explicitly shown in the bottom row of the diagram) and the standard algorithm.

Since we may then compare any equivalent series of moves to the algorithm, and this sense of equivalence is transitive, the algorithm provides a bridge between any two series of operations which result in the same KTGD. Therefore, the compositions of these relations constitute all the relations on **KTGD**.

As an example, the following figures show how to construct a relation between two different methods of constructing



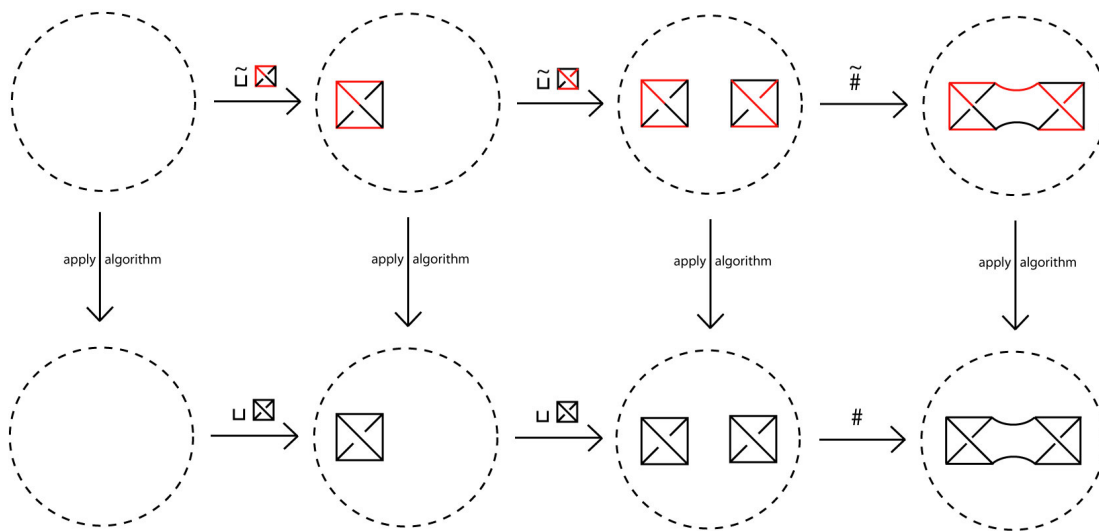


Figure 3.30: A relation showing that the algorithm on



is equivalent to two tetrahedron disjoint unions followed by a connect sum

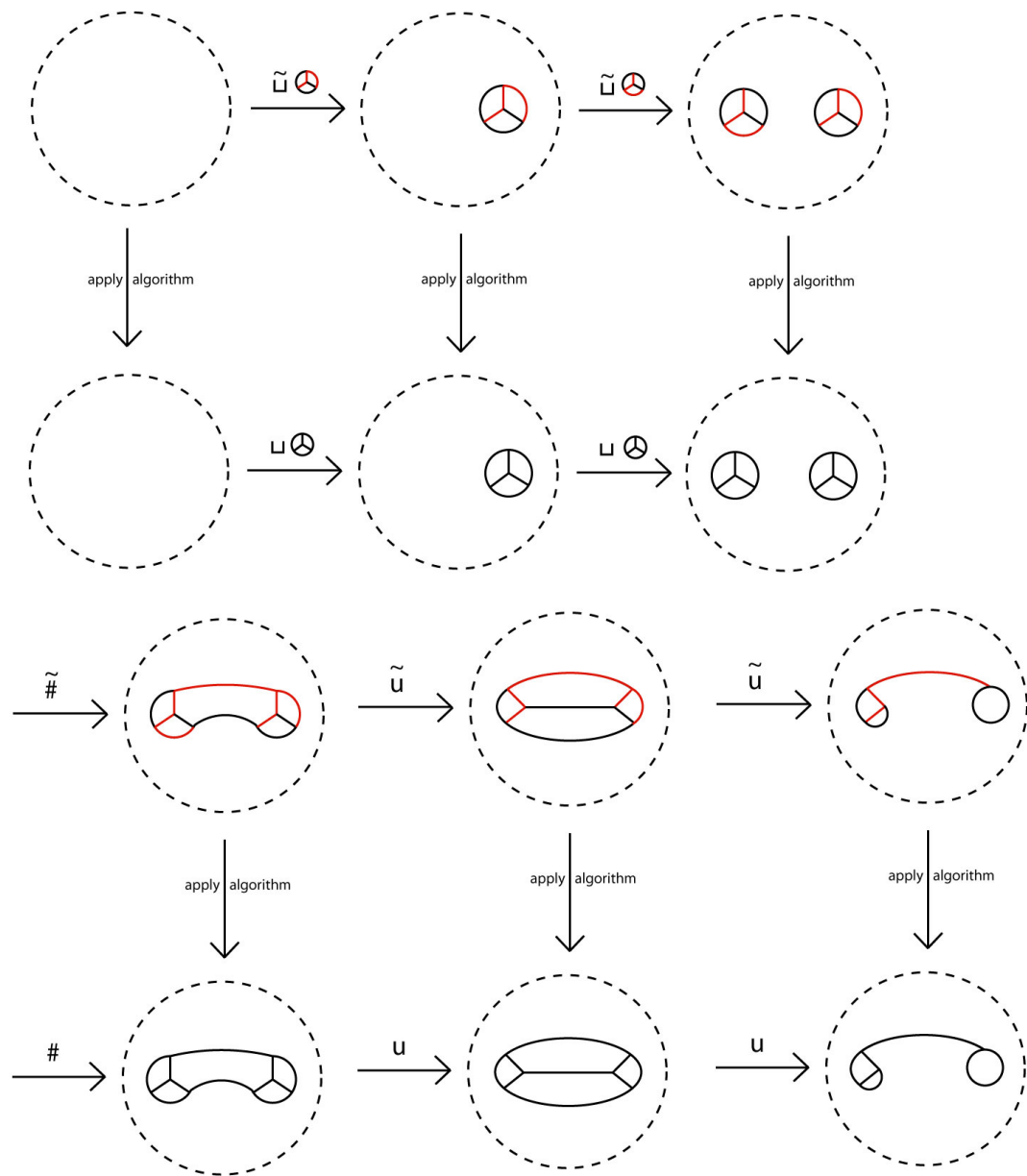


Figure 3.31: The composition of relations corresponding to a sequence of operations which generates



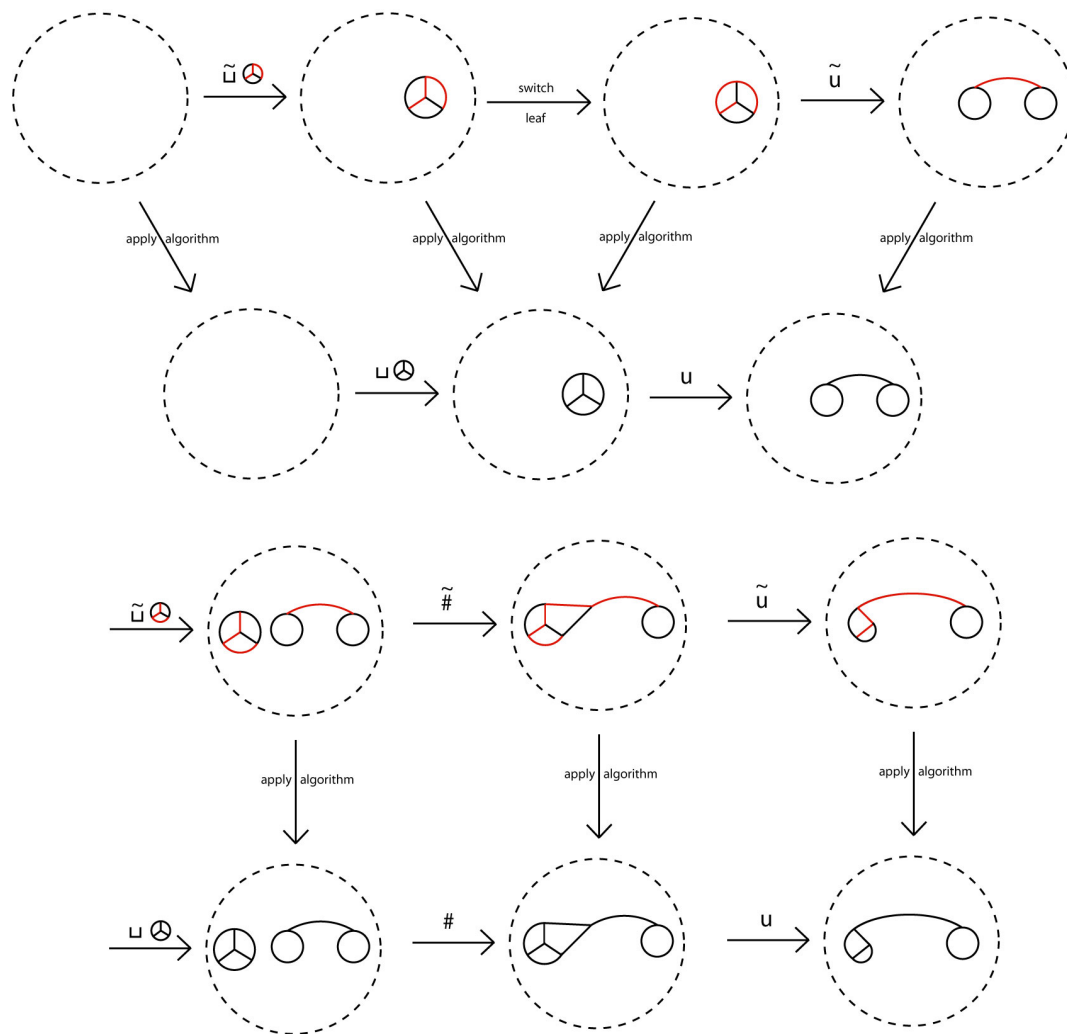


Figure 3.32: The composition of relations corresponding to another sequence of operations which generates



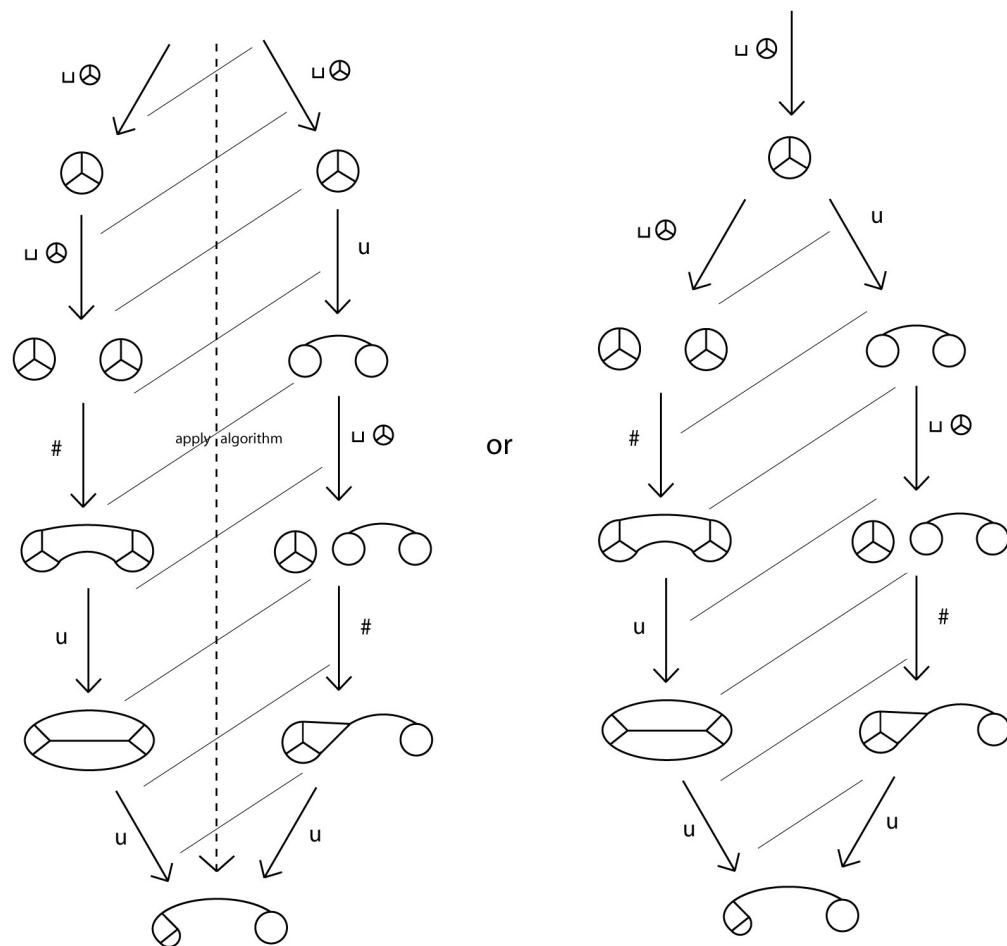


Figure 3.33: The relation obtained by noticing that both sequences above are equivalent to the algorithm

Chapter 4

Knotted trivalent graphs

We now begin the work of translating all this machinery to KTGs themselves. First, we must describe the behavior of the various operations on an object embedded in \mathbb{R}^3 . KTGs must be equipped with thickened edges to allow the basic operations to function on **KTG**, and we must define a set of KTGs with preferred edges. Additionally, there are many different KTGDs immersed in S^2 which correspond to the same KTG embedded in \mathbb{R}^3 ; we must define Reidemeister moves between two different diagrams of a trivalent graph and add more relations which correspond to these moves. In this case there will be five relevant moves rather than the three for knots [Kauffmaninvariants, Yamada].

4.1 Sets

The definition of knotted trivalent graphs given in Chapter 2 requires a small tweak before we can define their operations; specifically, since the previous definitions relied heavily on the planarity of the KTGD, we must provide thickenings for each edge which will indicate how twisted each edge is in order to have enough information to define our operation. We can think of each edge in this way as a ribbon rather than a strand. KTGs are certainly considered without such thickenings in the literature[] but our operations will require them.

The KTGDs we have been working with correspond to KTGs which are immersed in S^2 so that the ribbons are all planar. However, this is not always possible

for all KTGs, so we will depict twists on edges of KTGDs when necessary.

We will also need one more set:

Definition 6. The set of knotted trivalent graphs with thickened edges containing twist information and one preferred edge will be referred to as **KTGe**, and elements of this set as KTGes.

The extended list of Reidemeister moves for KTGs determined by Kaufmann [4] is provided in Chapter 1. Since we know that all these moves map to the identity on KTGDs we can make the following addition to our diagram of maps between sets:

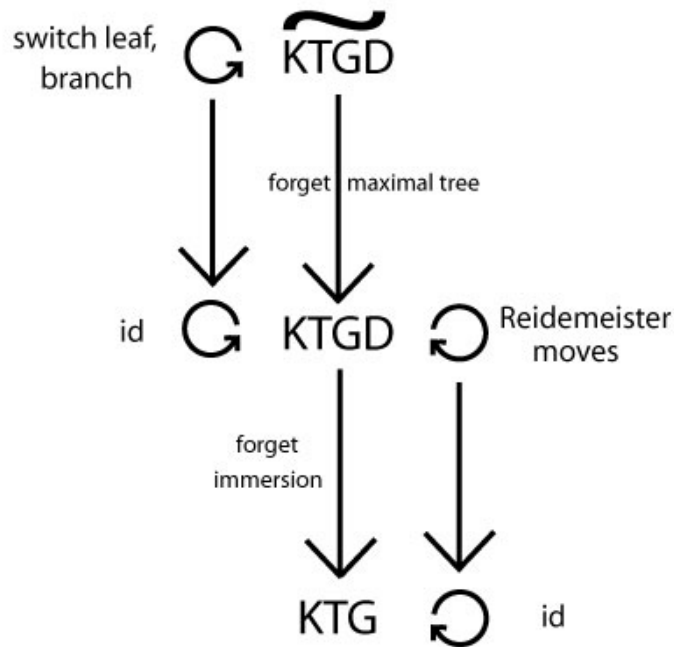


Figure 4.1: Maps between sets

4.2 Operations

We will now also consider Reidemeister moves which can operate on both **KTGD** and $\widetilde{\text{KTGD}}$ in the usual way. The only subtlety is that now that our objects are equipped with thickenings, we must be careful with twisting during moves 1 and 4.

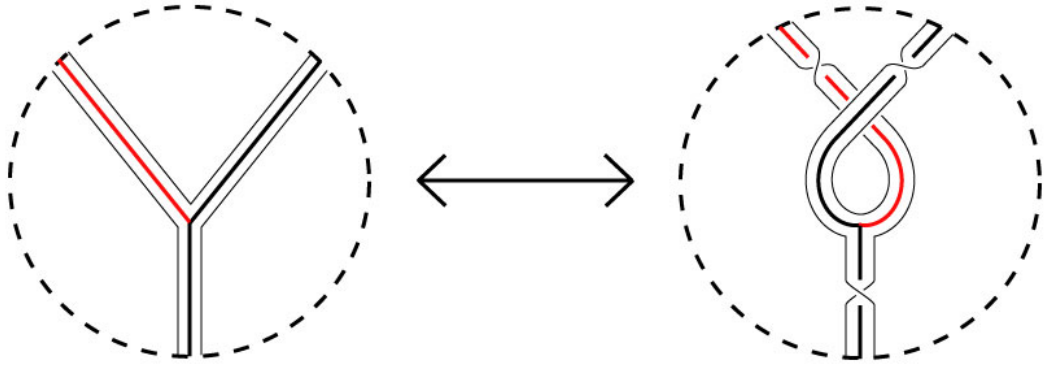


Figure 4.2: Reidemeister move 4 on a \widetilde{KTGD}

As with KTGDs, there are three basic operations on KTGs:

- Connect sum is a binary operation $\# : \mathbf{KTGe} \times \mathbf{KTGe} \longrightarrow \mathbf{KTG}$ where the selected KTGEs must be separate components and the selected edges must be exterior edges. Both arcs are cut and the ends are reattached to the ends from the other arc as indicated by the thickening. This operation is illustrated in Figure 3.2, where any portion of the KTGs which lies outside the dotted circle is unchanged. Note that the input arcs must be from two disjoint KTGs, or else the choice of how to splice them together is not well defined.

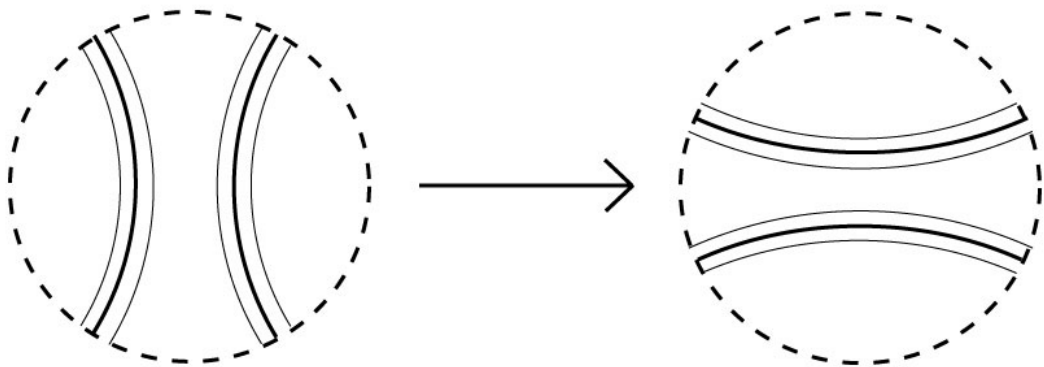


Figure 4.3: Connect sum

- Unzip is a unary operation $u : \mathbf{KTGe} \longrightarrow \mathbf{KTG}$ which splits an edge con-

necting two distinct vertices. Unzip removes an entire edge and its endpoints by connecting the other edges at those vertices to one another, again as indicated by the thickening. Unzip is illustrated in Figure 3.3, with areas outside the dotted circles remaining unchanged.

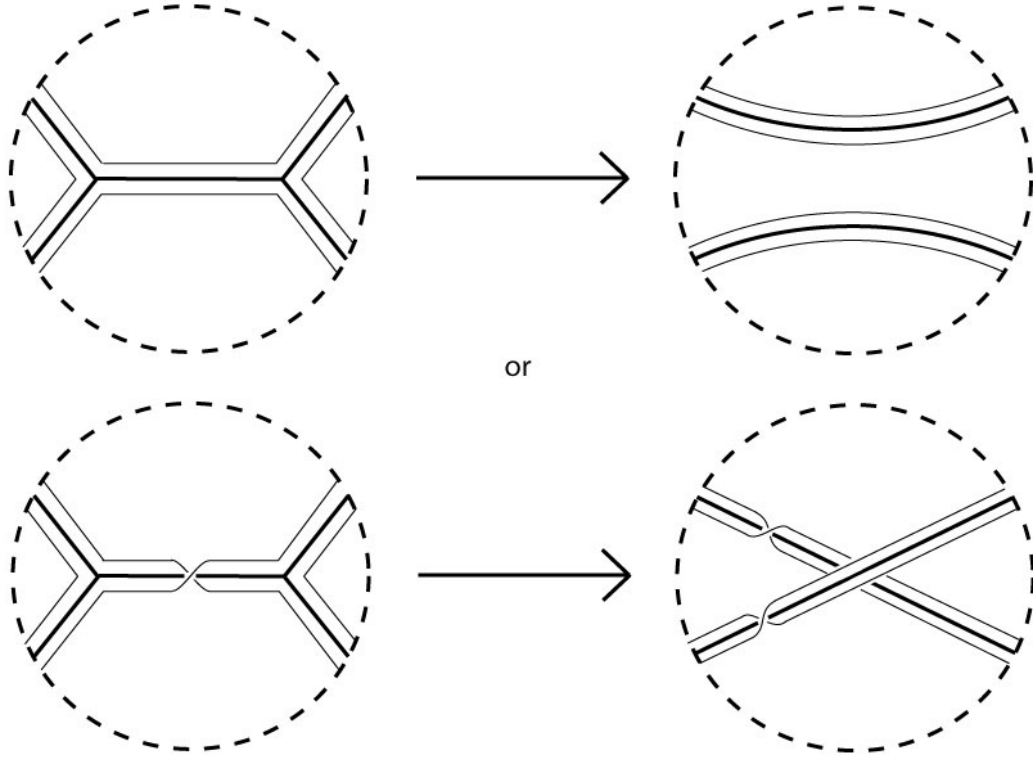


Figure 4.4: Unzip from two different views in \mathbb{R}^3

- Disjoint union is a binary operation $\sqcup : \mathbf{KTG} \times \mathbf{KTG} \longrightarrow \mathbf{KTG}$ which simply places two KTGs adjacent to one another in \mathbb{R}^3 .

4.3 Generators

Since KTGs with thickened edges may not be immersible in S^2 with all the ribbons planar, we will need two more generators to account for the twisting which some KTGDs will exhibit.

- Mobius strips

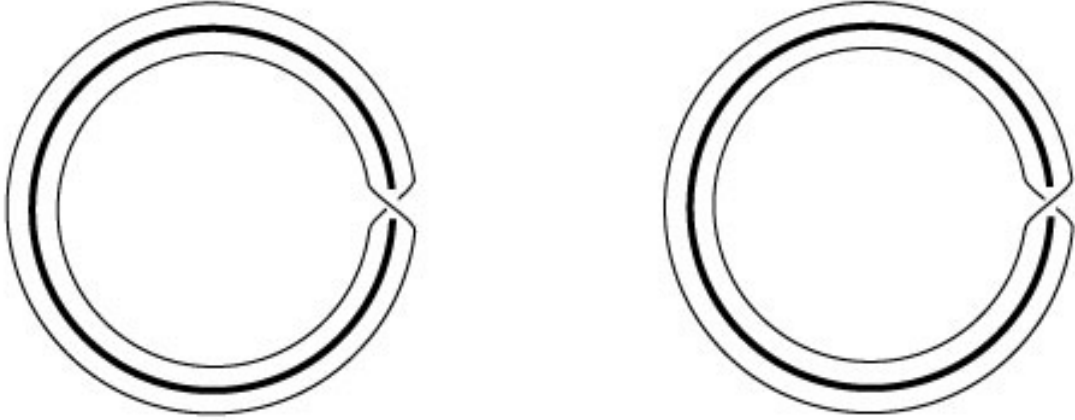


Figure 4.5: Two more generators needed for nonplanar KTGDs

The generating algorithm will follow exactly the same procedure for steps 0 through 4, and all twists in edges can be added in a final additional step.

4.4 Relations

In Chapter 3 we proved that the four basic relations outlined sufficed to generate all relations on **KTGD**; that is, for any two sequences of operations which result in identical KTGDs, a relation can be constructed by composing the tetrahedron, twisted tetrahedron, connect sum, and unzip relations. However, there are many different KTGDs which represent the same KTG, all related by our extended Reidemeister moves. As a result there will be more relations on **KTG** corresponding to these Reidemeister moves.

The following commutative diagrams illustrate the sequences of operations which yield equivalent KTGs; the two pictures in the $\widetilde{\mathbf{KTGD}}$ row represent different $\widetilde{\mathbf{KTGD}}$, and the two in the **KTGD** row do also, but since they correspond to the same KTG, following the arrows down the left side of the diagram or down the right yield identical results in the final **KTG** row.

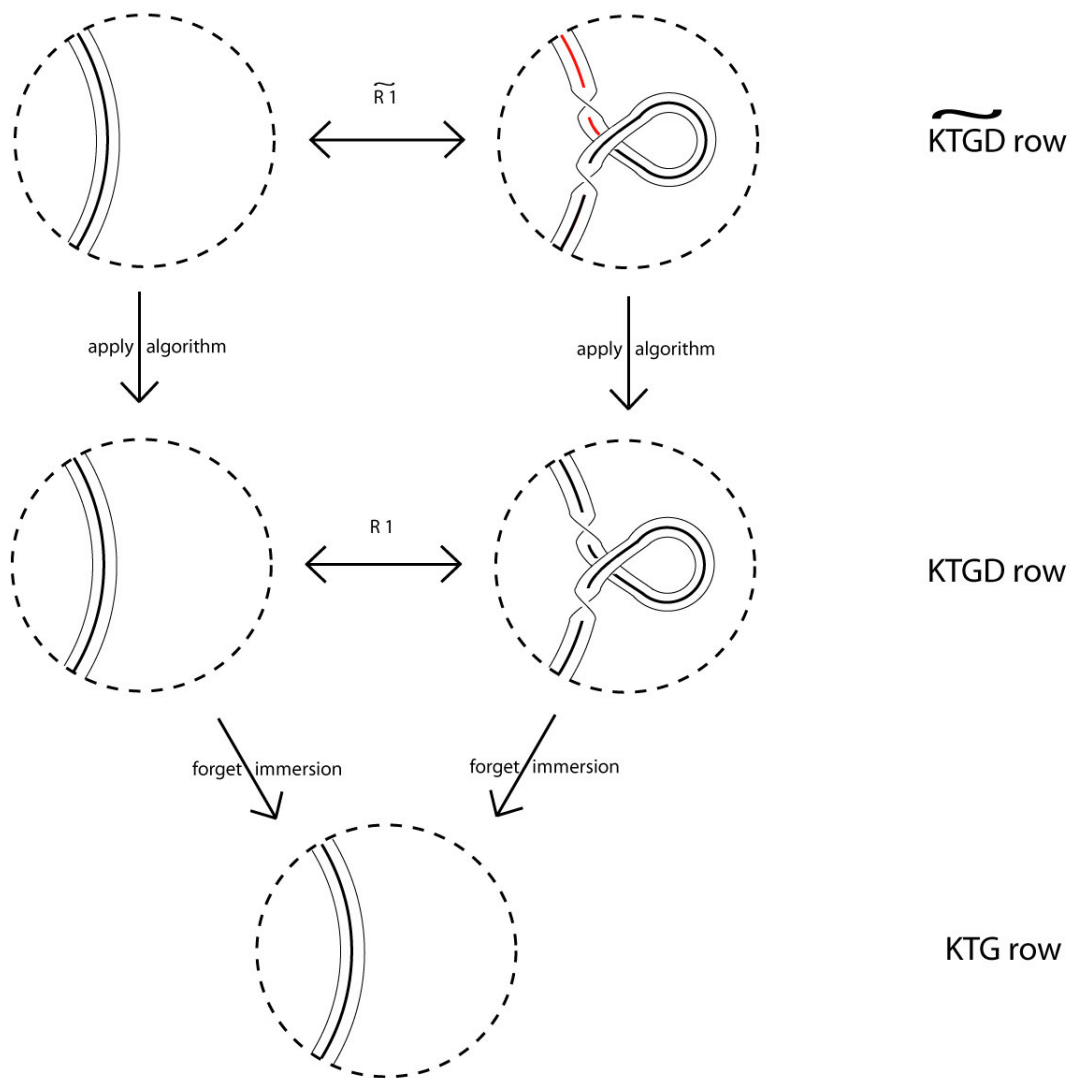


Figure 4.6: The relation corresponding to the Reidemeister 1 move

Theorem 3. (Kauffman): *Ambient isotopy of knotted trivalent graphs is generated by the moves shown in Figure (1.3). That is, any two diagrams of isotopic knotted trivalent graphs are related by a finite sequence of these moves.*

The proof of Theorem 3 employs the elementary combinatorial isotopies used by Reidemeister to prove the sufficiency of these extended Reidemeister moves to generate isotopy of knots. Details are given in [Kauffman, Yamada].

Corollary. *All Reidemeister moves on **KTGD** map to the identity on **KTG**, and*

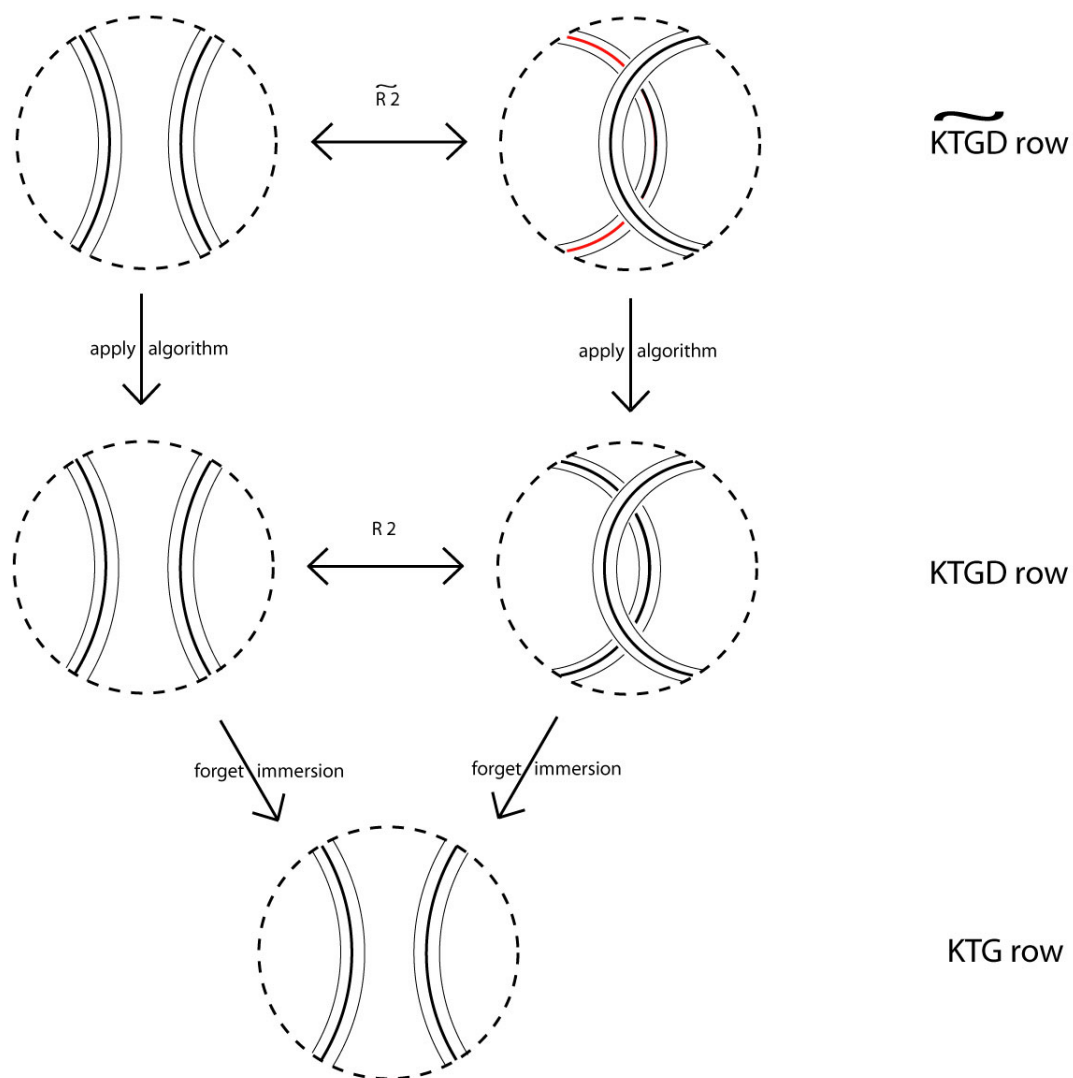


Figure 4.7: The relation corresponding to the Reidemeister 2 move

*the relations thus generated are the only ones required to extend our list of relations to **KTG***

For the compositions of these basic KTG relations which make up the pentagon relation, see the Appendix.

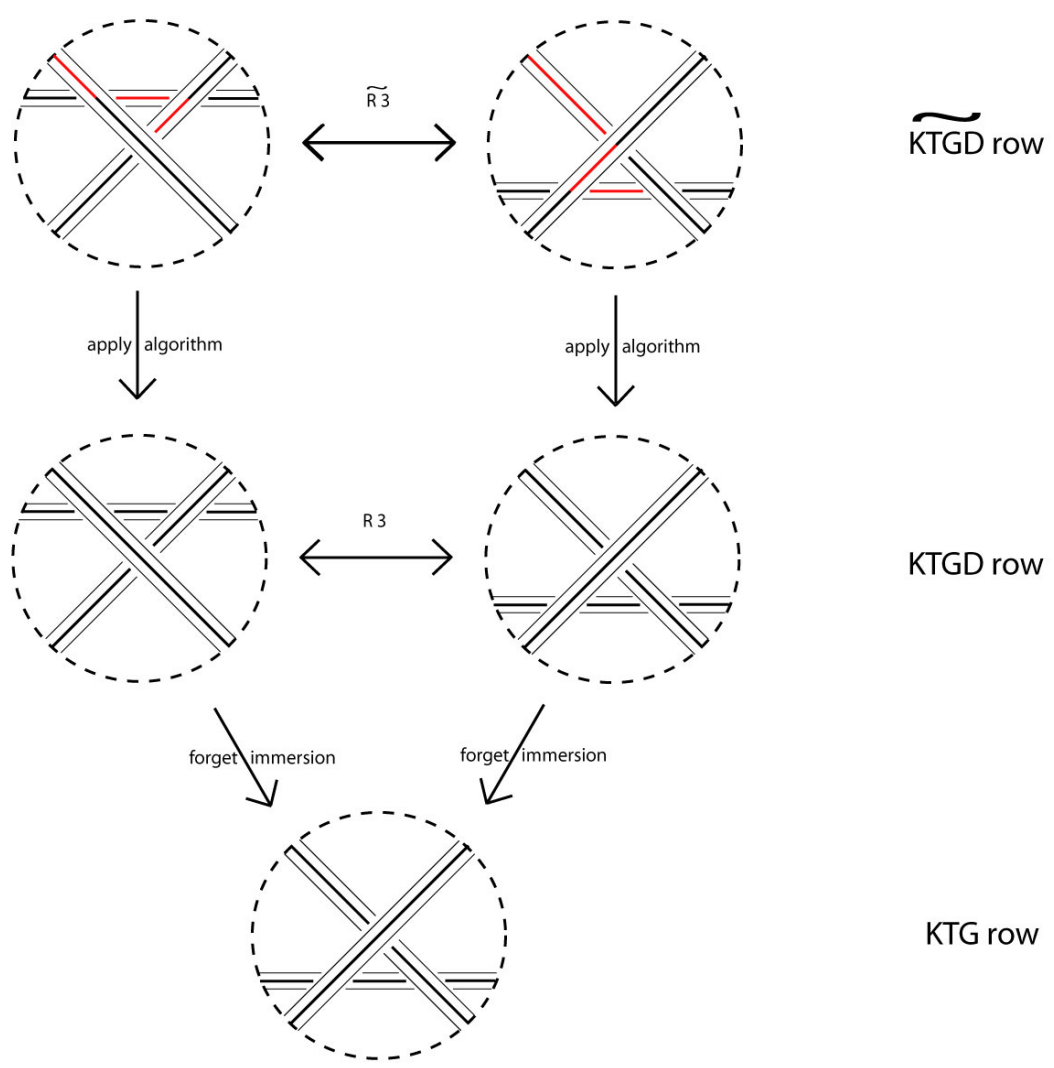


Figure 4.8: The relation corresponding to the Reidemeister 3 move

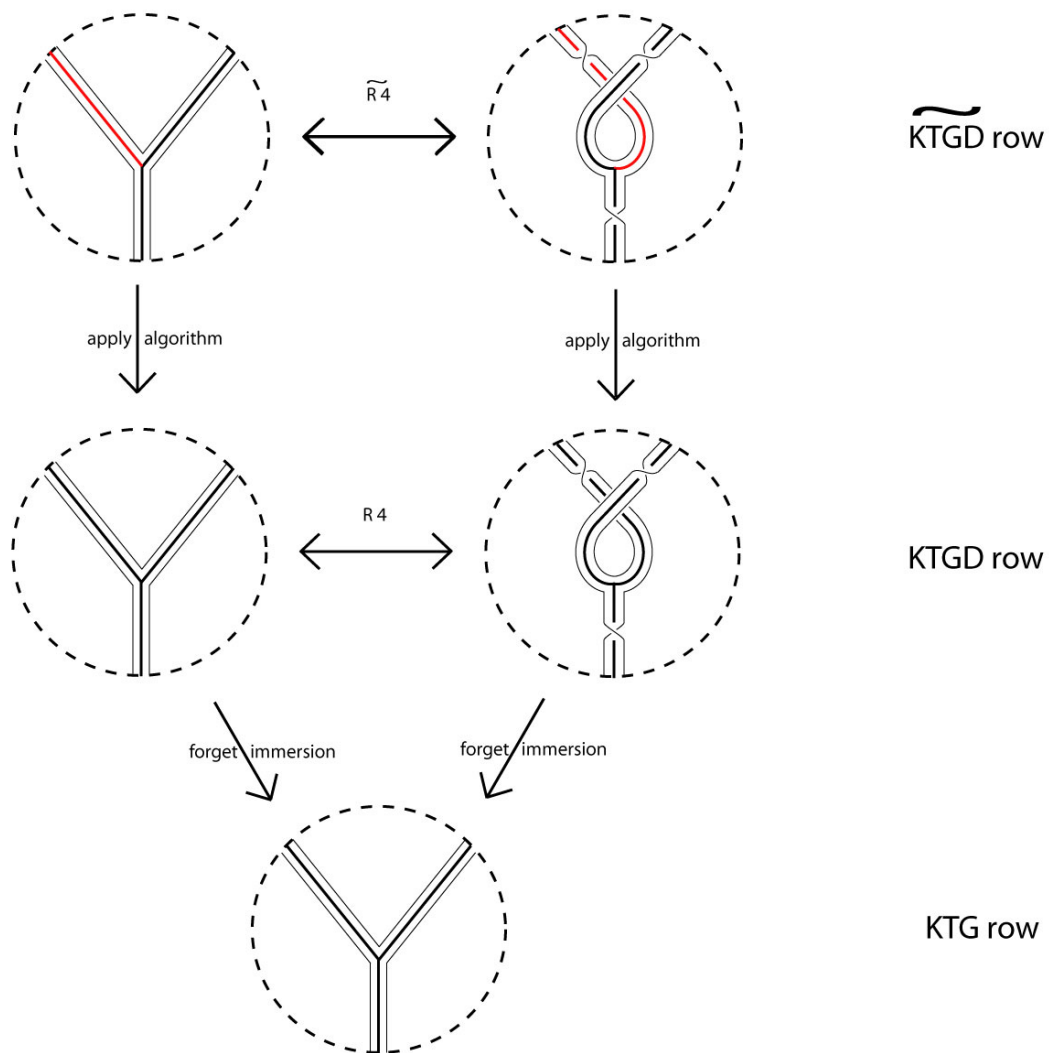


Figure 4.9: The relation corresponding to the Reidemeister 4 move

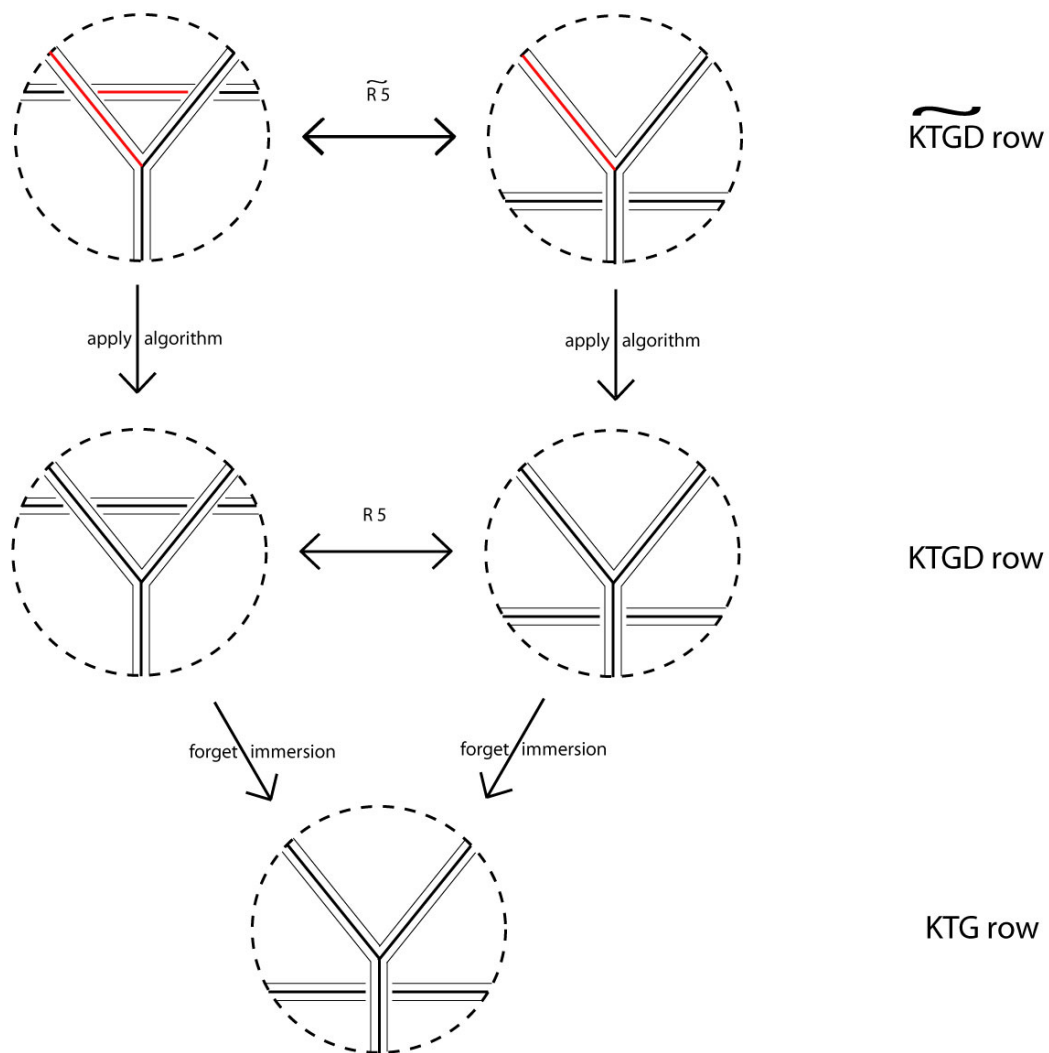


Figure 4.10: The relation corresponding to the Reidemeister 5 move

Appendix A

The pentagon relation

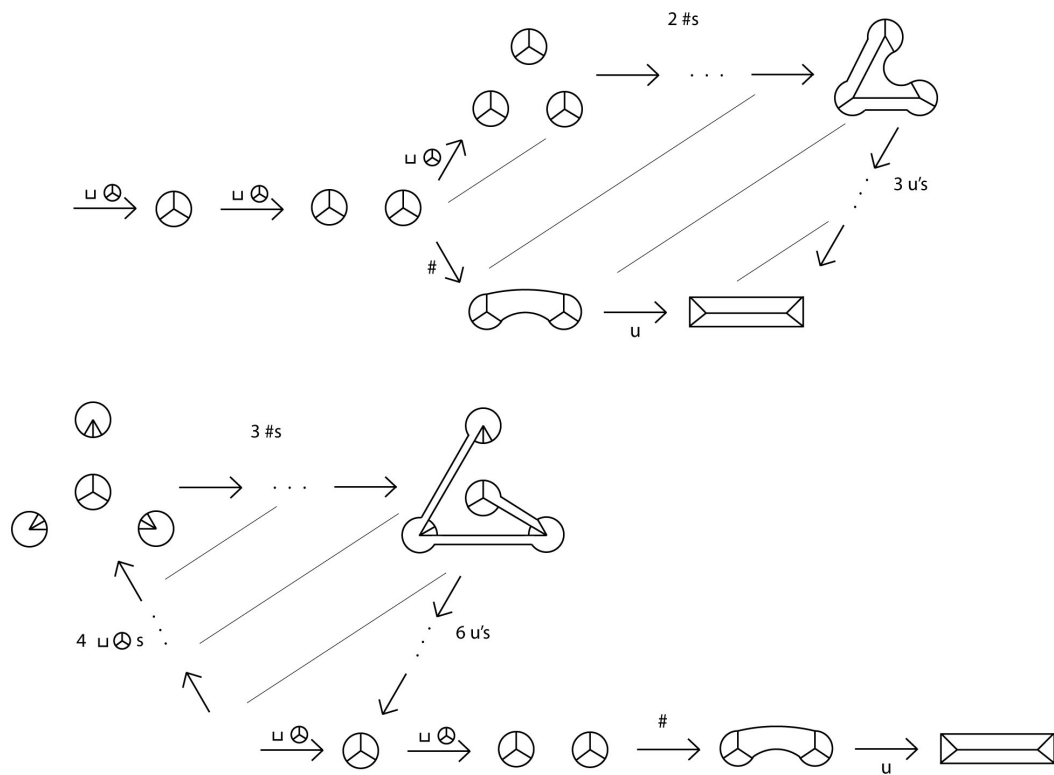


Figure A.1: The pentagon relation, with the formulation given in Thurston [8] above and a version which highlights its similarity to the tetrahedron relation below

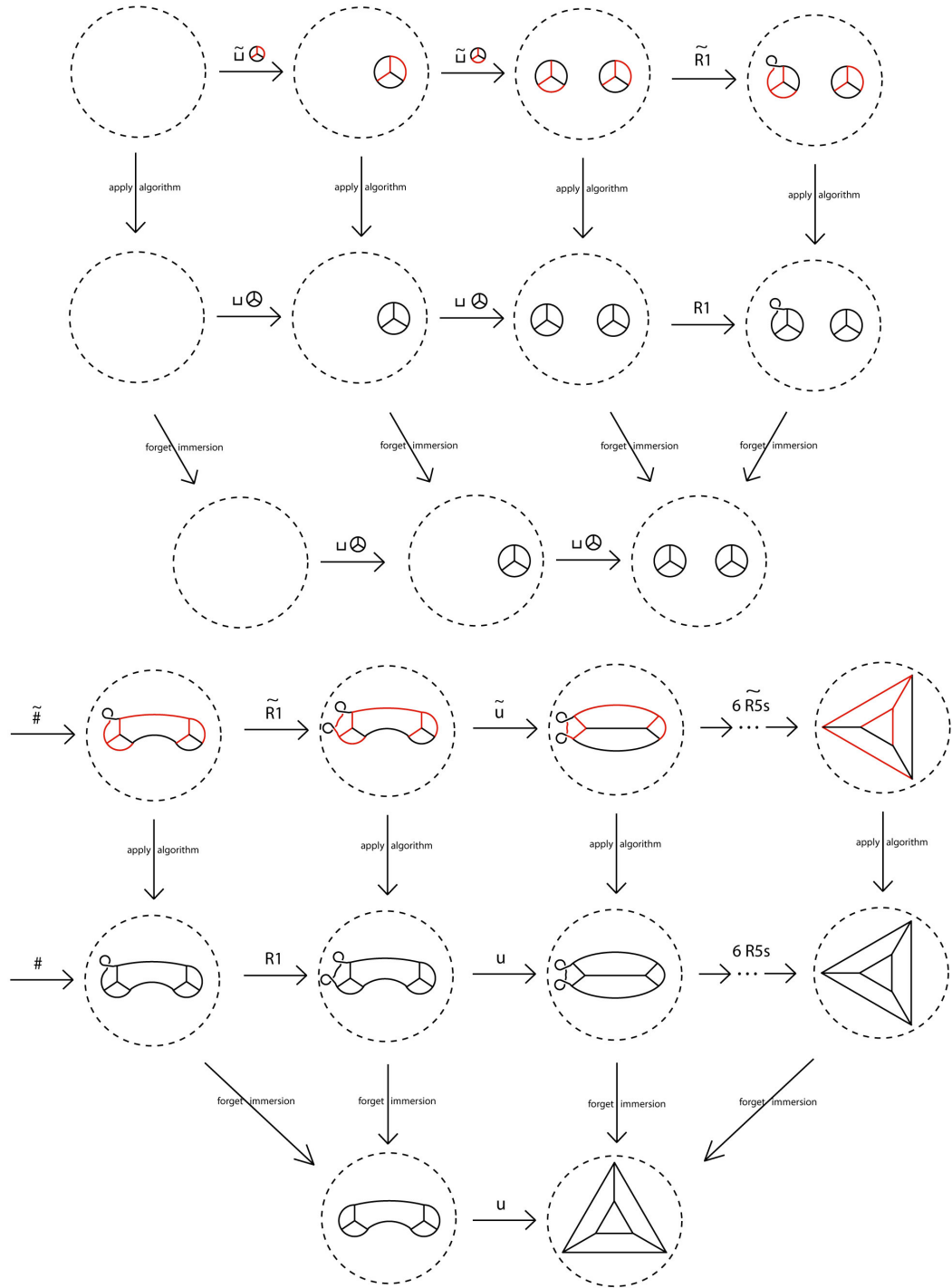


Figure A.2: The composition of relations for one half of the pentagon relation

Bibliography

- [1] J. Alexander and G. Briggs: *On types of knotted curves*, Ann. of Math. (2) **28** (1926/27), no. 1-4 562-586
- [2] D. Bar-Natan: *On the Vassiliev knot invariants*, Topology. **34**, 1995, 423-472
- [3] D. Bar-Natan, S. Garoufalidis, L. Rozasonsky, and D. Thurston: *The Århus integral of rational homology 3-spheres*, Selecta Math. (N. S.) **8** (2002), 315-339
- [4] L. Kauffman: *Invariants of Graphs in Three-Space* Trans. Amer. Math. Soc. **311** (1989) 697-710
- [5] M. Kontsevich: *Vassiliev's knot invariants*, Adv. in Sov. Math. **16**, part 2, 1993, 137-150
- [6] K. Reidemeister: *Elementaire Begrundung der Knotentheorie*, Abh. Math. Sem. Univ. Hamburg **5** (1926), 24-32
- [7] N. Reshetikhin and V. Turaev: *Invariants of 3-manifolds via link polynomials and quantum groups* Invent. Math. **103** (1991), 547-597
- [8] D. Thurston: *The algebra of knotted trivalent graphs and Turaev's shadow world* Geometry & Topology Monographs **4** (2002) 337-362
- [9] V. Turaev and O. Viro: *State sum invariants of 3-manifolds and quantum 6j-symbols*, Topology **31** (1992), 865-902
- [10] T. Watanabe: *Knotted trivalent graphs and the construction of the LMO invariant from triangulations* Osaka J. Math. **44** (2007), 351-362
- [11] S. Yamada: *An Invariant of Spatial Graphs* J. Graph Theory **13**, no. 5 (1989) 537-551