The multivariable Alexander polynomial (MVA) is a classical invariant of knots and links. We give an extension to regular virtual knots which has simple versions of many of the relations known to hold for the classical invariant.

By following the previous proofs that the MVA is of finite type we give a new definition for its weight system which can be computed as the determinant of a matrix created from local information. This is an improvement on previous definitions as it is directly computable (not defined recursively) and is computable in polynomial time. We also show that our extension to virtual knots is a finite type invariant of virtual knots.

We further explore how the multivariable Alexander polynomial takes local information and packages it together to form a global knot invariant, which leads us to an extension to tangles. To define this invariant we use so-called circuit algebras, an extension of planar algebras which are the ‘right’ setting to discuss virtual knots. Our tangle invariant is a circuit algebra morphism, and so behaves well under tangle operations and gives yet another definition for the Alexander polynomial. The MVA and the single variable Alexander polynomial are known to satisfy a number of relations, each of which has a proof relying on different approaches and techniques. Using our invariant we can give simple computational proofs of many of these relations, as well as an alternate proof that the MVA and our virtual extension are of finite type.
Acknowledgements

I must first of all thank my advisor, Dror Bar Natan, for his expert guidance and helpful encouragement during this project. Without his advice the math may never have been done and the thesis would not have been finished. The organization and appearance of this work has been greatly improved by his attention to detail.

My thanks go also to my fellow students for their friendship and support throughout this long process. I would especially like to thank everyone in our “knot theory group” for many valuable meetings and discussions.

I need to thank Sean Fitzpatrick who probably spent more time proofreading my thesis than he should have.

I would like to thank Charlie Frohman for his feedback on my thesis, and Ida Bulat for making this process run smoothly.

# Contents

1 Introduction
   1.1 Executive Summary .................................................. 1
   1.2 Introduction .......................................................... 2

2 Preliminaries
   2.1 The Reidemeister Moves ............................................... 4
   2.2 Virtual Knots and Links .............................................. 5
   2.3 Tangles ............................................................... 6
      2.3.1 Long Knots ...................................................... 7
   2.4 Singular Knots ....................................................... 8
   2.5 The algebra of chord diagrams ..................................... 9
   2.6 Arrow Diagrams ..................................................... 10
   2.7 Finite Type Invariants ............................................. 11
   2.8 Finite type invariants of virtual knots. ........................... 12
   2.9 Planar Algebras ................................................... 14
      2.9.1 Oriented Planar algebras .................................... 15

3 The Multivariable Alexander Polynomial
   3.1 The Alexander Matrix ............................................... 17
   3.2 The Normalization .................................................. 19
   3.3 The MVA for regular long virtual knots ......................... 22
Chapter 1

Introduction

1.1 Executive Summary

The multivariable Alexander polynomial (MVA) has many definitions, some of which are based on local information, such as which arcs are involved at each crossing. We use one such definition to define an extension, $v_{\text{MVA}}$, of the MVA to long regular virtual knots and links. This extension satisfies many simpler versions of relations that the MVA is known to satisfy.

We give a local definition for the weight systems of the MVA which improves on previous definitions as it is computable in polynomial time. We also show that our extension to long regular virtual knots is a finite type invariant of virtual knots (in the sense of [Pol00]) and give a definition for its weight system.

We work with so-called circuit algebras, an extension of planar algebras, and a nice setting in which to discuss virtual knots. We define a circuit algebra morphism, $t_{\text{MVA}}$, which is an extension of the MVA to tangles. This map commutes with circuit algebra maps, so we can glue together different tangles. This extension also gives us a new, simple proof that the MVA and our extension $v_{\text{MVA}}$ are finite type invariants. We can even use circuit algebra maps to define their weight systems on parts of chord and arrow
These extensions, which all commute with circuit algebra maps, provide an easy way to prove many of the relations satisfied by the MVA and its weight systems, since each relation can be verified by a simple calculation. Even better, we have written a Mathematica program that does the calculation for us.

1.2 Introduction

What is a knot? A knot is a ‘nice’ embedding of the unit circle ($S^1$) into the 3-sphere ($S^3$) or sometimes $\mathbb{R}^3$, up to ambient isotopies. We can define a ‘nice’ embedding in many different ways (piecewise linear, or $C^\infty$) so as to avoid pathological examples known as wild knots (shown in Figure 1.1). We also need to take these embeddings up to ambient isotopy since we do not want to allow a knot to be pulled so tight that it disappears, as depicted in Figure 1.1. Once we have formally defined knots, they are exactly what you would want them to be: thin, infinitely elastic bands in 3-space. We will also work with $n$-component links which are nice embeddings of $n$ copies of $S_1$.

These are not ambient isotopic.

A wild (i.e. not ‘nice’) knot.

Figure 1.1: What is a knot?

The most basic question in knot theory is “How can you tell two knots apart?” This is a deceptively difficult question. The earliest known knot invariants were easy to define, like the unknotting number or the minimal crossing number, but very difficult to
compute. In the early 20th century a common approach was to look at the knot group (the fundamental group of the knot complement); however, this reduces the problem to the equally difficult group isomorphism problem (How do you know if two presentations define the same group?).

In 1928, using the knot group Alexander [Ale28] defined the first easy to compute knot invariant, now known as the Alexander polynomial (or multivariable Alexander polynomial on links). Alexander’s invariant was a polynomial defined up to signs and powers of the variables, which made it easy to compare the results for different knots. Alexander verified that his polynomial separates all knots with less than 8 crossings. Though originally defined only up to signs and powers, various renomalizations have been given to make this an honest polynomial invariant. In 1967 Conway [Con70] gave a set of relations which define the Conway potential function, which is really the MVA in disguise.

In his paper on Vassiliev invariants [BN95] Bar-Natan showed that all of the knot polynomials are of finite type. Dynnikov [Dyn97] and H. Murakami [Mur99] showed that the MVA is of finite type, and H. Murakami gave a set of relations which define its weight system. We give a new definition of its weight system which improves on H. Murakami’s as it is computable in polynomial time.

In 1997 Kauffman introduced virtual knots [Kau99], an extension of classical knot theory by adding a new type of crossing. Many classical knot invariants including the single variable Alexander polynomial have been extended to this new class of knots. In Section 3.3 we give an extension of the MVA to long regular virtual knots. This extension can be shown to satisfy many relations related to relations that the classical MVA satisfies.
Chapter 2

Preliminaries

2.1 The Reidemeister Moves

The work of Reidemeister in 1926 [Rei26] showed that the ambient isotopy equivalence classes of knots in 3-space are equivalent to planar projections of knots under the three Reidemeister moves in Figure 2.1.

![Reidemeister Moves Diagram]

Figure 2.1: The Reidemeister Moves

At times we will consider the set of framed knots. These are knots defined as above, with the added information of a continuous choice of normal vector, called a framing. These are considered up to ambient isotopies in the space of framed knots, so they can be deformed, but we must continuously change the framing. As before, we can represent these with planar diagrams, with blackboard framing. The equivalence of framed knots is given by planar isotopy and Reidemeister 2 and 3, together with a modified version of the Reidemeister 1 move.
Definition 2.1. Two knots diagrams are said to be regularly isotopic if they are equivalent via R2 and R3. We call knot diagrams up to R2 and R3 equivalence regular knots.

Orientation

We will often require that our knots, or links, or tangles be oriented, which means that we have a preferred direction to travel along each path. We indicate orientation with an arrow. Oriented diagrams have two types of crossings, positive and negative, as shown in Figure 2.3. When working with oriented knots there are oriented versions of the Reidemeister moves.

2.2 Virtual Knots and Links

In 1997 Kauffman created a new ‘virtual’ crossing and defined a new type of knot theory: virtual knot theory [Kau99]. Virtual knots are represented by diagrams with positive, negative and virtual crossings. A so-called virtual crossing is represented, as in Figure 2.4 by intersecting arcs. These diagrams are taken up to equivalence under the classical Reidemeister moves, as well as the following virtual Reidemeister moves, and one mixed
Chapter 2. Preliminaries

Reidemeister move:

![Reidemeister Moves Diagram]

Figure 2.4: The virtual and mixed Reidemeister moves.

You may notice that there is only one ‘mixed Reidemeister 3’ move (with both real and virtual crossings). The other two possibilities shown in Figure 2.5 are known as Forbidden Moves 1 and 2, and are, as the name suggests, forbidden. In fact, if you allow both forbidden moves, you can unknot all classical knots [GPV00].

![Forbidden Moves Diagram]

Figure 2.5: The Forbidden moves.

These virtual knot diagrams have a topological interpretation as knots in thickened surfaces modulo stabilization [Kau99].

2.3 Tangles

We can think of tangles as being parts of a knot or link. Intuitively, a tangle is the intersection of a knot with a closed 3-ball; the formal definition is as follows:
**Definition 2.2.** A tangle (or n-tangle) is a ‘nice’ embedding of n copies of the unit interval I and some number (possibly 0) copies of the unit circle into $\mathbb{B}^3$, such that $\partial I \subset \partial \mathbb{B}^3$. Tangles are considered up to boundary fixing ambient isotopies.

As before, using Reidemeister’s Theorem we can consider tangles as projected onto the plane. Thus, tangles are diagrams drawn in circles with fixed end points modulo the Reidemeister moves and planar isotopy.

![Figure 2.6: A 1-tangle and a 2-tangle.](image)

**Virtual Tangles**

We can consider virtual tangles as simply a generalization of classical tangles, with the addition of virtual crossings and the virtual Reidemeister moves.

### 2.3.1 Long Knots

**Definition 2.3.** A long knot is a knot which has one ‘long’ cut arc.

![Figure 2.7: A long trefoil.](image)

We can think of long knots (or links) as either a 1-tangle, or as a knot with one strand that passes though a point at infinity. In the case of classical knots and links, there is a 1-1 correspondence between closed knots and long knots; however this is no longer true for virtual knots and links [Kau99].
2.4 Singular Knots

Definition 2.4. [BN95] A knot (or link) with a singular point represents the formal linear combination of the knots which differ at only one crossing, one with the singular point resolved as a positive crossing, the other as a negative crossing.

\[ \begin{array}{c}
\text{\includegraphics[width=0.2\textwidth]{singular_point_resolution}} \\
\end{array} \]

Figure 2.8: The resolution of a singular point.

Definition 2.5. A singular knot with \( m \) singular points represents the formal linear combination of \( 2^m \) knots.

Example 2.6.

\[ \begin{array}{c}
\text{\includegraphics[width=0.8\textwidth]{singular_knot_sum}} \\
\end{array} \]

Figure 2.9: This singular knot represents the formal sum of these four knots.

Notation. We will let \( K_m \) be the space of singular knots with \( m \) singular points. Thus \( K_0 \), for example, is the space of classical knots.

Every singular knot diagram has an underlying chord diagram which indicates the ordering of the singular points along the knot. We can think of this as the preimage of the knot with a chord connecting the two points which are mapped to the same singular point in the embedding of the singular knot. An \( n \) component singular link will have an underlying chord diagram with chords on \( n \) circles.

Example 2.7.
To find the underlying chord diagram for a singular link, we may begin by numbering the singular points.

Then on a second set of circles (one for each link component) we record which singular points we meet as we walk around the corresponding link component.

Finally we connect the two points which correspond to the same singular point with a chord.

### 2.5 The algebra of chord diagrams

**Definition 2.8.** A chord diagram is a collection of circles (called the skeleton) with some number of chords drawn on them.

We will use solid arcs to denote the skeleton components and dotted lines to denote the chords.

**Definition 2.9.** The algebra of chord diagrams is formal linear combinations of chord diagrams modulo the four term relation (4T) shown in Figure 2.10.

![Figure 2.10: The 4T relation.](image-url)
Figure 2.10 represents the sum of four chord diagrams which differ only at the sites drawn. We impose the 4T relation due to the topology of singular knots [BN95].

When we work with chord diagrams we often introduce trivalent vertices, to represent the following difference of chord diagrams. Notice that because of the 4T relation the three ways of resolving the trivalent vertex are equivalent.

\[
\begin{align*}
&= \\
\end{align*}
\]

Figure 2.11: This relation is sometimes called STU.

\section{2.6 Arrow Diagrams}

Arrow diagrams are an oriented version of chord diagrams. We will use them when working with finite type invariants of virtual knots.

\textbf{Definition 2.10.} An arrow diagram is a collection of circles (the skeleton) with some number of oriented chords (arrows).

As in the case of chord diagrams we will draw the skeleton with solid lines and the arrows with dotted lines. On arrow diagrams we impose a six term relation (an oriented version of 4T) [Pol00], it is shown in Figure 2.12.

\[
\begin{align*}
&= \\
\end{align*}
\]

Figure 2.12: The 6T relation.

We allow trivalent vertices on arrow diagrams as a short hand for the difference of the two arrow diagrams indicated below. We do not allow sources or sinks (all arrow
pointing in or all arrows pointing out of a vertex). These relations are also known as ‘STU’ relations.

![Figure 2.13: The STU relations for arrow diagrams.](image)

**2.7 Finite Type Invariants**

Every knot invariant defines an invariant of singular knots, by extending it linearly to the signed sum of knots which the singular knot represents. An invariant that vanishes on knots with many singular points is called finite type:

**Definition 2.11.** [BN95] If $\phi : K \to G$ where $G$ is any abelian group, we say that $\phi$ is a finite type invariant of degree $m$ if it vanishes on all knots with $m + 1$ singular points.

In other words, an invariant of type $m$ is zero on $K_n$ for all $n > m$ (ie. knots with more than $m$ singular points). For knots in $K_m$ a finite type invariant of type $m$ depends only on the underlying chord diagram not the topological embedding of the actual singular knot [BN95]. We call the map associated to an invariant on the level of chord diagrams its weight system.

**Example 2.12.** The Conway Polynomial [Con70]
Chapter 2. Preliminaries

The Conway polynomial is closely related to the single variable Alexander polynomial and can be defined by the following rules:

\[ \nabla \left( \begin{array}{c} \rightarrow \\ \leftarrow \end{array} \right) - \nabla \left( \begin{array}{c} \leftarrow \\ \rightarrow \end{array} \right) := z \ \nabla \left( \begin{array}{c} \leftarrow \\ \leftarrow \end{array} \right) \]

\[ \nabla \left( \begin{array}{c} \text{k circles} \\ \end{array} \right) := \begin{cases} 1 & \text{if } k = 1 \\ 0 & \text{if } k \neq 1 \end{cases} \]

The coefficient of \( z^m \) in the Conway polynomial is a type \( m \) invariant \([BN95]\) since using the definition, any singular point can be smoothed at the price of a factor of \( z \). Diagrammatically,

\[ \nabla \left( \begin{array}{c} \rightarrow \\ \leftarrow \end{array} \right) := \nabla \left( \begin{array}{c} \leftarrow \\ \rightarrow \end{array} \right) - \nabla \left( \begin{array}{c} \leftarrow \\ \leftarrow \end{array} \right) = z \ \nabla \left( \begin{array}{c} \leftarrow \\ \leftarrow \end{array} \right). \]

So if there are \( m + 1 \) singular points, smoothing all of them will contribute a factor of \( z^{m+1} \); hence, the coefficient of \( z^m \) will be 0. So the coefficient of \( z^m \) is a type \( m \) invariant.

Using a similar argument, if we expand the Alexander polynomial using the substitution \( x = e^t = 1 + t + \frac{t^2}{2} + \ldots \), and let \( \Delta_n(K) \) be the term of order \( n \) in the expansion, i.e.;

\[ \Delta(K) = \sum_{n=0}^{\infty} \Delta_n(K). \]

One can check that each of the \( \Delta_n \)'s are of finite type \( n \).

2.8 Finite type invariants of virtual knots.

When working with virtual knots and links there are two ways to define finite type invariants. We can consider finite type invariants in the same way as we do on classical knots, where a singular knot with a double point represents the difference between a positive and a negative crossing, and then finite type invariants are given by weight
systems on chord diagrams. We will use a different concept of finite type invariants for virtual knots introduced in [GPV00]. Instead of working with double points we will work with knots with semi-virtual crossings, this leads to arrow diagrams.

**Definition 2.13.** A semi-virtual crossing as shown below, represents the formal linear combination of the two (virtual) knots indicated.

\[
\begin{array}{c}
\begin{tikzpicture}
  \draw[thick] (0,0) -- (0,1);
  \draw[thick] (0,1) -- (1,1);
  \draw[thick] (1,1) -- (1,0);
  \draw[thick] (1,0) -- (0,0);
\end{tikzpicture}
\end{array} := \begin{tikzpicture}
  \draw[thick] (0,0) -- (0,1);
  \draw[thick] (0,1) -- (1,1);
  \draw[thick] (1,1) -- (1,0);
\end{tikzpicture} - \begin{tikzpicture}
  \draw[thick] (0,0) -- (0,1);
  \draw[thick] (0,1) -- (1,1);
  \draw[thick] (1,1) -- (1,0);
\end{tikzpicture}
\]

Note that when working with oriented knots there are two types of semi-virtual crossings.

\[
\begin{array}{c}
\begin{tikzpicture}
  \draw[thick] (0,0) -- (0,1);
  \draw[thick] (0,1) -- (1,1);
  \draw[thick] (1,1) -- (1,0);
\end{tikzpicture} = \begin{tikzpicture}
  \draw[thick] (0,0) -- (0,1);
  \draw[thick] (0,1) -- (1,1);
  \draw[thick] (1,1) -- (1,0);
\end{tikzpicture} - \begin{tikzpicture}
  \draw[thick] (0,0) -- (0,1);
  \draw[thick] (0,1) -- (1,1);
  \draw[thick] (1,1) -- (1,0);
\end{tikzpicture}
\quad \text{and} \quad
\begin{tikzpicture}
  \draw[thick] (0,0) -- (0,1);
  \draw[thick] (0,1) -- (1,1);
  \draw[thick] (1,1) -- (1,0);
\end{tikzpicture} = \begin{tikzpicture}
  \draw[thick] (0,0) -- (0,1);
  \draw[thick] (0,1) -- (1,1);
  \draw[thick] (1,1) -- (1,0);
\end{tikzpicture} - \begin{tikzpicture}
  \draw[thick] (0,0) -- (0,1);
  \draw[thick] (0,1) -- (1,1);
  \draw[thick] (1,1) -- (1,0);
\end{tikzpicture}
\]

Figure 2.14: A positive semi-virtual crossing and a negative semi-virtual crossing.

**Definition 2.14.** An invariant is of finite type \(m\) if it vanishes on all knots with more than \(m\) semi-virtual crossings.

One can show that for an invariant \(\phi\) of type \(m\), if we restrict to knots with \(m\) semi-virtual crossings, \(\phi(\mathcal{K}) = -\phi(\mathcal{K})\) [GPV00]. As in the classical case \(\phi\) will not depend on the embedding of the knot, but on the ordering of the semi-virtual crossings along the skeleton of the knot. So once we know that an invariant is of type \(m\) since \(\phi(\mathcal{K}) = -\phi(\mathcal{K})\), we only need to define the weight system for knots with all \(\mathcal{K}\) type crossings. We represent the pre-image of a semi-virtual \(\mathcal{K}\) crossing as an arrow, where the arrow starts on the overstrand and goes to the understrand.

Finite type invariants of virtual knots are determined by weight systems on arrow diagrams modulo the 6T relation [GPV00].
Figure 2.15: A link with only semi-virtual over crossings and its underlying arrow diagram.

2.9 Planar Algebras

Jones introduced planar algebras in the study of subfactors [Jon99], which lead to the definition of the Jones polynomial. The language of planar algebras is a particularly nice way to describe tangles and many tangle and knot invariants.

**Definition 2.15.** A planar tangle with $n$ inputs is a diagram with no crossings drawn in a large disk with $n$ smaller disks removed. There is a marked point along each of the boundary circles.

Figure 2.16: A planar tangle with two inputs.

**Definition 2.16.** A planar algebra is a collection of vector spaces $V$ and morphisms $P$ between these spaces, where the vector spaces $V_n$ are indexed by the natural numbers and the maps are indexed by planar tangles.
For example, the tangles below represent a particular map $\rho : V_4 \otimes V_6 \to V_8$. Both represent the same map since such tangles are taken up to boundary fixing planar isotopy.

There is a natural composition of planar tangles, we require that the maps that they represent commute with composition. Note that the marked points tell you how to fit one tangle inside the other.

2.9.1 Oriented Planar algebras

If we add an orientation to the arcs, each component will have either an incoming arrow (indexed by $+$) or an outgoing arrow (indexed by $-\,$). The spaces are now indexed by ordered lists of $+$ and $-$, which we diagrammatically can represent as arrow pointing into and out of a disk, and the diagrams have oriented arcs. The orientation on the diagrams must match the spaces that it is applied to, and to compose two such maps the orientations must match. Every disk will have a marked point and we will order the arrows by going around the circle in the counterclockwise direction starting at the marked point.

Example 2.17. Let $V$ be any vector space and let each incoming arrow correspond to an copy of $V$, and each outgoing arrow correspond to a copy of $V^*$. If you take the pairing map to be interior multiplication then, this is a planar algebra. However there is
Figure 2.17: An oriented planar algebra map from $V_{\{-,+,+,+\}} \otimes V_{\{+,+-,-\}}$ to $V_{\{+,+,+,-\}}$.

No need to restrict to only planar connections, we can also use this as an example of a circuit algebra.

**Example 2.18.** Tangles form a planar algebra, where the objects are tangles and the maps are gluing the tangles to a planar tangle. Note that the space of classical tangles is generated by the positive and negative crossing. If you add virtual crossings you get virtual tangles. However if we are working with virtual tangles there is no need to restrict ourselves to planar gluings, so we introduce the concept of circuit algebras.
Chapter 3

The Multivariable Alexander Polynomial

In 1928 Alexander used the Dehn presentation for the knot group to define the first knot polynomial, now called the multivariable Alexander polynomial (MVA) \cite{Ale28}. His original definition was defined only up to sign and powers of the variables. We will follow the approach in \cite{Tor53} and use a definition that comes from the Wirtinger presentation, which has the advantage of having a normalization to fix the ambiguities of signs and powers.

3.1 The Alexander Matrix

We first construct a so-called Alexander matrix for the link; for more information about Alexander matrices and their properties see \cite{Rol76}. We start by labeling all of the arcs in a diagram for our link. We can extend this to a labeling of the crossings, so that the \( i \)th arc starts at the \( i \)th crossing. We have labeled the diagram in Figure 3.1 following these rules. The variables in the MVA come from associating a variable to each link component.
The matrix we construct will have one row for each crossing, and one column for each arc. The entries in the $c^{th}$ row are determined by the rule in Figure 3.2, where $x$ and $y$ are the variables associated to the participating link components. All of the other entries in the $c^{th}$ row are zero. This matrix, $M(D)$, is an Alexander matrix for the link [Tor53].

Example 3.1. For the link shown in Figure 3.1 we would construct the following Alexander matrix.

$$
M(D) = \begin{array}{cccccc}
  & a_1 & a_2 & a_3 & a_4 & a_5 \\
 a_1 & y - 1 & 0 & 0 & 0 & 1 - x \\
 a_2 & 1 - y & x & 0 & 0 & -1 \\
 a_3 & -y & 0 & 1 & y - 1 & 0 \\
 a_4 & 0 & 0 & -y & 1 & y - 1 \\
 a_5 & 0 & 0 & y - 1 & -y & 1 \\
\end{array}
$$
Since both the rows and the columns have the same labels we require that we list the rows in the same order that we list the columns. We denote by $M(D)_{ij}$ the Alexander matrix $M(D)$ with the $i^{th}$ row and $j^{th}$ column of the matrix removed.

We can now define $MVA_{original}$, Alexander’s original multivariable polynomial.

**Definition 3.2.** $MVA_{original} := \frac{\det(M(D)_{ij})}{t_j - 1}$

It is a consequence of the linear dependence of the columns that we must divide by $t_j - 1$; for a proof see [Ale28].

Like Alexander’s original definition this is not yet a knot invariant, since it depends on the choice of the deleted row and the choice of knot diagram. In the following section we will define a few easily computed functions which will eliminate the ambiguities due to these choices.

### 3.2 The Normalization

To eliminate the sign ambiguity, we only need to add a factor of $(-1)^{i+j}$. The rows in the matrix are linearly dependent, and the choice of which row to delete will change the determinant by a monomial. To cancel this factor for the crossing corresponding to the row we will be deleting, we must find an escape path.

**Definition 3.3.** An escape path is a path starting from a point to the right of both outgoing strands of a crossing, to the unbounded region of the plane. From the path we will record the monomial $w_i$ that we get by following the path, recording the variable associated to the arcs we cross: $x_j$ if the arc is crossing the path from left to right, and $x_j^{-1}$ if it is crossing from right to left.

**Example 3.4.** For the following link, we have indicated an escape path from the 5th crossing; by following the path we see that $w_5 = y^2$. 

Multiplying by this monomial will eliminate the ambiguity due to the choice of the deleted row. For an explanation of why this factor is needed see [Har83].

With the factors already defined we have an invariant of link diagrams. However the Reidemeister 1 and 2 moves will introduce additional powers of the variables. To eliminate these changes we introduce the following notation.

**Definition 3.5.** We let $\mu(k)$ be the number of times the $k^{th}$ component is the over strand in a crossing.

We let $\text{rot}(k)$ be the rotation number of the $k^{th}$ link component (i.e. how many times the tangent vector to the $k^{th}$ component rotates in the counterclockwise direction as we travel around the component once).

The factor $\prod_k x_k^{\frac{\text{rot}(k)-\mu(k)}{2}}$ will eliminate the ambiguities due to the Reidemeister moves. We can now give a definition of the multivariable Alexander polynomial.

**Definition 3.6.** [Tor53] The multivariable Alexander polynomial (MVA) of a link $L$ is denoted by $\Delta(L)$ and is given by,

$$\Delta(L) = (-1)^{i+j} \det(M(L)^j_i) \left( \prod_k x_k^{\frac{\text{rot}(k)-\mu(k)}{2}} \right),$$

where $w_i$ is an escape path as in Definition 3.3, $M(L)$ is the Alexander matrix defined in Section 3.4, and $\text{rot}(k)$ and $\mu(k)$ are as in Definition 3.5 [Tor53].

One can prove that $\Delta$ is a link invariant by verifying that it is unchanged by the Reidemeister moves, and does not depend on the row and column deleted; this is shown...
in [Tor53]. Our calculation in Section 8.2.1 contains the bulk of the proof invariance under the Reidemeister moves.

Note that on knots this differs from the usual definition of the single variable Alexander polynomial by a factor of \((x - 1)\). On links \(x_j - 1\) divides \(|M_j^i|\), so that \(\Delta(L)\) is indeed a (Laurent) polynomial [Tor53].

**Example 3.7.** For the following labeled link, we construct the given matrix.

\[
\begin{array}{cccccc}
  a_1 & a_2 & a_3 & a_4 & a_5 \\
  a_1 & y-1 & 0 & 0 & 0 & 1-x \\
  a_2 & 1-y & x & 0 & 0 & -1 \\
  a_3 & -y & 0 & 1 & y-1 & 0 \\
  a_4 & 0 & 0 & -y & 1 & y-1 \\
  a_5 & 0 & 0 & y-1 & -y & 1 \\
\end{array}
\]

We will be deleting the 5th row which corresponds to the circled crossing. The escape path is the dotted path, which begins at a point to the right 5th crossing; by following it we see that \(w_5 = y^2\). Checking that \(\mu(1) = 1, \mu(2) = 4, \text{rot}(1) = -1, \text{rot}(2) = -2\), and \(M_5^5 = x(y - 1)(1 - y + y^2)\) gives the MVA of the link as \(y^{-1}(1 - y + y^2)\).

**Remark 3.8.** In later computations we will may want to divide an arc at a non-crossing, meaning that we wish to label half of the arc \(a\) and the other half \(b\). We can do this by adding the following row to the Alexander matrix.

\[
\begin{array}{cccc}
  a & b \\
  a & 1 & -1 \\
\end{array}
\]

**Lemma 3.9.** Dividing an arc will not change value of \(\Delta\) computed using Definition 3.6.
Proof. Suppose we have a knot diagram, and we label it two different ways, which differ only on one arc. In the first labeling we label that arc by $c$, in the other we label the first half of that arc by $b$ and the second half by $a$. When we construct the Alexander matrix for both labellings, the matrices will be almost the same, having the form shown below.

\[
\begin{pmatrix}
C & M \\
\end{pmatrix} \quad a 
\begin{pmatrix}
1 & -1 \\
A & B \\
\end{pmatrix} 
\rightarrow 
\begin{pmatrix}
1 & 0 \\
A & B \\
\end{pmatrix}
\]

Note that the row labeled $c$ in the first matrix will be the same as the row labeled $b$ in the second matrix. The sum of the columns $A + B$ will be equal to $C$, since the contributions from the arcs $a$ and $b$ must equal those from $c$. But then by the above column operation we can change the second matrix into one which must have the same determinant as the first.

\[\square\]

3.3 The MVA for regular long virtual knots

It is not possible to define a normalization which is invariant under both virtual and classical R1, so we choose to work with regular isotopy classes of virtual knots (invariant only under R2 and R3). For virtual knots, the minors of the Alexander matrix $M(K)$ constructed in Section 3.1 are no longer multiples of each other, so we need to specify which row to delete. We do this by working with long virtual knots, as defined in Section 2.3.1 and always deleting the column corresponding to the incoming long arc. We also remove $rot(k)$ and the $\omega_i$ from the previous normalization, as they were needed only for R1 invariance. We should note that unlike the classical case, long virtual knots are not in 1-1 correspondence with closed virtual knots [Kau99].

Note that since the incoming long arc does not start at a crossing, it will not have a corresponding row in the Alexander matrix. When we label the arcs virtual crossings do
not divide the arcs.

**Definition 3.10.** The multivariable Alexander polynomial for long virtual knots and links \((vMA)\) is:

\[
\Delta_v(L) = \frac{\det(M(L))}{x_i - 1} \prod_k x_k^{\mu(k)} ,
\]

where \(M(L)\) is the matrix defined as in Section 3.6, \(\bar{M}(L)\) is the minor obtained by deleting the column corresponding to the incoming long arc, \(\mu(k)\) is the number of times the \(k^{th}\) component is the over strand in a crossing, and \(x_i\) is the variable associated to the long arc.

If you wish to consider virtual crossings as actually dividing arcs you may add the two rows shown in Figure 3.3 for every virtual crossing. Note that this is simply using Definition 3.8 twice to artificially divide the pair of arcs at this point.

![Figure 3.3: A rule for virtual crossings](image)

**Theorem 3.11.** \(\Delta_v\) is an invariant of virtual knots and links.

**Proof.** The proof that this is invariant under the classical Reidemeister moves is identical to that proof for the MVA on classical knots and appears in other sources such as [Ale28].

The verification for the virtual and mixed Reidemeister moves is trivial, since if you chose not to divide the arcs at virtual crossings, the matrices produced will be identical. If you choose to divide arcs at virtual crossings, then you are simply using Definition 3.8 multiple times, and we know it does not change final determinant. 

\[\square\]
**Example 3.12.** We have labeled all of the arcs in this long virtual knot; notice that we chose to divide the left arc into $g$ and $f$. Since $a$ is the label of the incoming long arc we will be deleting the $a^{th}$ column.

![Diagram of a long virtual knot with labeled arcs.]

When we delete the $a^{th}$ column, the determinant is $(x-1)(1-2x+2x^2)y$. We note that $\mu(1) = 3$, $\mu(2) = 1$ so the normalization factor will be $x^{-2}y^{1/2}/(x-1)$, giving a value of $\Delta_v(K) = \frac{1 - 2x + 2x^2}{x^2 \sqrt{y}}$.

In Chapter 7 we will give many relations which this invariant satisfies. These are simpler versions of the relations which the classical MVA was known to satisfy. We will also see that this is true of the weight systems derived from these invariants.
Chapter 4

Finite Type Invariants and Weight Systems

Murakami [Mur99] and Dynnikov [Dyn97] used Torres’ [Tor53] normalization of the MVA to show that under an appropriate change of variables the MVA is of finite type. By following Murakami’s proof we will give a new definition for the weight system of the MVA which is computable in polynomial time. We will also show that the extension vMVA is a finite type invariant of virtual knots.

4.1 The Multivariable Alexander Polynomial is a Finite Type Invariant

Theorem 4.1. [Mur99, Dyn97]

If we denote by $\text{MVA}_n(K)$ the terms of order $n$ in the expansion of the MVA after the substitution $x_i \mapsto e^{x_i} = 1 + t_i + \frac{t_i^2}{2!} + \ldots$, i.e.

$$\Delta(K) = \sum_{n=0}^{\infty} \text{MVA}_n(K),$$

\footnote{For historical reasons we use this substitution [BN95], however any substitution of the form $x_1 \mapsto 1 + t_i + O(2)$ would work.}
then the terms $MVA_n$ are finite type invariants of order $n + 1$.

**Proof.** Recall from Definition 2.11 that an invariant is of finite type $n$ if it vanishes on knots with more than $n$ singular points (singular knots are defined in Definition 2.4). To show that $MVA_n(K)$ is a finite type invariant, we will show that each singular point adds 1 to the degree of the lowest degree term in the MVA after a particular substitution.

As shown in Definition 2.4 a knot with a singular point is really just the difference between the two knots. Let us label the crossings from a singular point as in the diagram below. We have divided each of the over arcs in to two, so that we may label both diagrams in the same way.

\[ a \quad b \quad c \quad d \]
\[ x \quad y \quad 1 - x \quad -1 \quad 0 \]
\[ b \quad 0 \quad 1 \quad 0 \quad -1 \]

The two Alexander matrices constructed following the rule in Figure 3.2 will differ in only the $a^{th}$ and $b^{th}$ rows, as shown below.

\[
\begin{pmatrix}
    a & b & c & d \\
    a & y & 1 - x & -1 & 0 \\
    b & 0 & 1 & 0 & -1
\end{pmatrix} -
\begin{pmatrix}
    a & b & c & d \\
    a & 1 & 0 & -1 & 0 \\
    b & y - 1 & 1 & 0 & -x
\end{pmatrix}
\]

When we wish to compute the MVA we will need to take the determinants of the above matrices, and multiply by a normalization factor. The only difference in the normalization coefficient for these two knots comes from the number of over crossings, which adds an extra factor of $x^{-\frac{1}{2}}$ to the right hand side and $y^{-\frac{1}{2}}$ to the other. By including the difference in normalization in the matrix, we get the following difference of determinants:
which we can simplify as follows, using row operations:

\[
\begin{vmatrix}
  y & 1-x & -1 & 0 & 0 \\
  0 & y^{-\frac{1}{2}} & 0 & -y^{\frac{1}{2}} & 0 \\
\end{vmatrix}
- \begin{vmatrix}
  x^{\frac{1}{2}} & 0 & -x^{\frac{1}{2}} & 0 & 0 \\
  y^1 & 1 & 0 & -x & 0 \\
\end{vmatrix}
\]

Using the multilinearity of the determinant we can see that the difference of determinants becomes:

\[
\begin{vmatrix}
  y & 1 & -1 & -x & 0 \\
  0 & y^{\frac{1}{2}} & 0 & -y^{\frac{1}{2}} & 0 \\
\end{vmatrix}
- \begin{vmatrix}
  x^{\frac{1}{2}} & 0 & -x^{\frac{1}{2}} & 0 & 0 \\
  y^1 & 1 & -1 & -x & 0 \\
\end{vmatrix}
\]

\[
\begin{vmatrix}
  y & 1 & -1 & -x & 0 \\
  0 & y^{\frac{1}{2}} & 0 & -y^{\frac{1}{2}} & 0 \\
\end{vmatrix}
+ \begin{vmatrix}
  x^{\frac{1}{2}} & 0 & -x^{\frac{1}{2}} & 0 & 0 \\
  y^1 & 1 & -1 & -x & 0 \\
\end{vmatrix}
\]

We can then write the determinant as follows:

\[
\begin{vmatrix}
  y & 1 & -1 & -x & 0 \\
  x^{\frac{1}{2}} & y^{\frac{1}{2}} & -x^{-\frac{1}{2}} & -y^{\frac{1}{2}} & 0 \\
\end{vmatrix}
\]

If we use the substitution \( e^{t_1} = x \), \( e^{t_2} = y \) and expand \( e^{t_1} \) as a power series, omitting terms of order greater than 1, we get

\[
\begin{vmatrix}
  \frac{1}{2}t_2 + \ldots & -\frac{1}{2}t_1 + \ldots & \frac{1}{2}t_2 + \ldots & \frac{1}{2}t_1 + \ldots & 0 \\
  1 - \frac{1}{2}t_1 + \ldots & 1 - \frac{1}{2}t_2 + \ldots & -1 + \frac{1}{2}t_1 + \ldots & -1 + \frac{1}{2}t_2 + \ldots & 0 \\
\end{vmatrix}
\]
The first row contains only elements of degree one or higher, so each singular point will add at least one to the lowest degree appearing in the determinant. If a knot has \( m \) singular points we can repeat the above manipulation to express the invariant as the determinant of a matrix which has at least \( m \) rows of degree 1. When we apply the same change of variables to the normalization coefficient, we notice that dividing by \( x_i - 1 \) lowers the exponent of \( t_i \) by one. Since the rest of the coefficient has leading term 1, the resulting term has degree at least \( m - 1 \). Hence for a link with \( m + 1 \) singular points the lowest degree term in the MVA is of degree at least \( m \). Therefore the term of degree \( m \) in the MVA under the given substitution is a finite type invariant of type \( m + 1 \).

We define a function, \( \text{cMVA} \), from chord diagrams on \( n \) circles to \( \mathbb{Z}[t_1, ..., t_n] \), which is the weight system of the MVA. Given a chord diagram and a variable associated to each component, we choose a marked point on one of the arcs. This divides that arc into two; we then label all of the arcs, as we do in Figure 4.1.

![Figure 4.1: A labeled chord diagram with a marked point.](image)

We now construct a matrix \( M(D) \) whose rows and columns are indexed by the arcs, with entries as follows.

As before, \( M_i^i(D) \) is the matrix \( M(D) \) with the \( i^{th} \) row and column removed.

**Definition 4.2.** [Arc08] The cMVA weight of a chord diagram \( D \) is

\[
\text{cMVA}(D) := \frac{\det(M_i^i(D))}{t_i},
\]

where \( i \) is the marked point.
Theorem 4.3. $c_{MVA}$ is the weight system for the term of order $n$ in the expansion of the multivariable Alexander polynomial under the substitution $x_i = e^{t_i} = 1 + t_i + \ldots$

A proof will follow this example.

Example 4.4. Using the labeled diagram in Figure 4.1, we construct a matrix indexed by its arcs. Here is one of the chords and the corresponding matrix entries.

We can use the rest of the chords to fill in the remainder of the table as follows (omitting the zeros):
To calculate the weight of this diagram, we simply remove the marked row and column \((a_1)\), take the determinant, and divide by \(t_1\) to get \(t_2^2\).

As in the proof of Theorem 4.1, we will be using the substitution \(x_i = e^{t_i} = 1 + t_i + \ldots\). Since we are interested in the terms of order \(m - 1\), and we know that the rows corresponding to each of the \(m\) double points will contribute a row of elements with order 1, we take a row of order 1 in the rows corresponding to the double points, and terms of order zero elsewhere. We note that up to order 0 the rows corresponding to positive and negative crossings are the same after this substitution. We will create a matrix with the following rows:

\[
\begin{array}{cccccc}
 1 & -1 \\
-\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\
-2_t & -2_t & t_1 & t_1 \\
t_2 & t_2 & -2_t & -2_t \\
t_2 & t_2 & -2_t & -2_t \\
-\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\
-\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2}
\end{array}
\]

The entries in the matrix for the classical crossings are identical to the entries in the matrix used in Lemma 3.9 to artificially divide an arc. Using Lemma 3.9 backwards we
can relabel the arcs so that \( x_{1,n-1} \) and \( x_{1,n} \) have the same label, then compute a new matrix with the same determinant. With the new labeling all the arcs will either begin at a singular point or at the marked point. If you replace the singular points with chords this is the matrix described in Theorem 4.3.

Let us now consider the normalization factor \( \frac{w_i}{t_i - 1} \prod_{k} t_k^{rot(k) - \mu(k)} \). Under the substitution the numerator becomes \( 1 + O(1) \), and the denominator is \( t_a + \ldots \). Therefore as we are interested in terms of order \( m - 1 \) the effect of the normalization is to divide by \( t_a \). This is the map described in Theorem 4.3.

In Section 6.4.1 we will give another definition for this weight system.

Recall from Section 2.8 that an invariant is a finite type invariant of virtual knots if it vanishes on knots with many semivirtual crossings. Similar to the case for classical knots, if we use a particular substitution the terms of order \( n \) in the expansion of the vMVA are of type \( n + 1 \).

**Theorem 4.5.** Under the substitution \( x_i \mapsto e^{t_i} = 1 + t_i + \frac{t_i^2}{2!} + \ldots \) the term of order \( m \) in the expansion of the vMVA is a finite type invariant of order \( m - 1 \) on long virtual knots.

**Proof.** To prove this theorem we must show that under the given substitution both types of semi-virtual crossings contribute terms of order one (or higher) to the vMVA. We will show the computation for the difference of a positive and a virtual crossing below (\( \setminus \)). The proof of the second case is equivalent.

\[
\begin{align*}
&\begin{array}{cccc}
c & y & b \\
x & d & a \\
\end{array} \\
\begin{array}{cccc}
a & b & c & d \\
1 & 0 & -1 & 0 \\
1 - x & y & 0 & -1 \\
\end{array}
- \\
\begin{array}{cccc}
a & b & c & d \\
a & 1 & 0 & -1 \\
b & 0 & 1 & 0 \\
-1 & \\
\end{array}
\end{align*}
\]

As before the normalization coefficient will differ slightly for each of these knots;
that is, the \( y \) strand will be the over crossing an extra time on the left hand side. We include the difference in the normalization coefficient \( \left( \frac{1}{\sqrt{y}} \right) \) in the first matrix. So when we compute the difference of the vMVAs of the knots, we will be taking the difference of the following two determinants.

\[
\begin{vmatrix}
1 & 0 & -1 & 0 & 0 \\
\frac{1-x}{\sqrt{y}} & \sqrt{y} & 0 & -\frac{1}{\sqrt{y}} & 0 \\
* & * & & & \\
\end{vmatrix}
- 
\begin{vmatrix}
1 & 0 & -1 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 \\
* & * & & & \\
\end{vmatrix}
\]

By the multilinearity of determinants the difference of the above two determinants is the same as the determinant shown below,

\[
\begin{vmatrix}
1 & 0 & -1 & 0 & 0 \\
\frac{1-x}{\sqrt{y}} & \sqrt{y} - 1 & 0 & 1 - \frac{1}{\sqrt{y}} & 0 \\
* & * & & & \\
\end{vmatrix}
\]

which under the substitution \( x_i \mapsto e^{t_i} = 1 + t_i + \frac{t_i^2}{2} + \ldots \) is:

\[
\begin{vmatrix}
1 & 0 & -1 & 0 & 0 \\
-t_1 + \ldots & \frac{1}{2}t_2 + \ldots & 0 & \frac{1}{2}t_2 + \ldots & 0 \\
* & * & & & \\
\end{vmatrix}
\]

A similar computation shows that the second type of semi-virtual crossing (\( \times \)) contributes a row of order one. So every semi-virtual crossing will increase the order of the lowest term in the expansion of the vMVA by one. Since the lowest order term for a single knot is of order \( -1 \), if there are \( m \) singular points the MVA will begin with at least order \( m - 1 \).

\[\square\]

Since we have proved that the vMVA is a finite type invariant of virtual knots, as shown in \[GPV00\] we may define its weight system as a map on arrow diagrams. As in the case of the MVA on classical knots we can define its weight system by taking
the determinant of a particular matrix. In this case given an arrow diagram we choose a
marked point at which to divide one arc. We then create a matrix with rows and columns
indexed by the arcs, where each arrow tells us how to fill in the rows corresponding the
exiting two arcs, by following the rule in figure 4.3.

$$
\begin{pmatrix}
1 & 0 \\
-1 & 0 \\
\end{pmatrix}
\begin{pmatrix}
1 & 0 & -1 & 0 \\
-t_1 & \frac{1}{2}t_2 & 0 & \frac{1}{2}t_2 \\
\end{pmatrix}
\begin{pmatrix}
a & b & c & d \\
a_1 & a_2 \\
a_1 & 1 & -1 \\
\end{pmatrix}
$$

Figure 4.3: A rule to create a matrix from an arrow diagram.

If we let \( \widehat{M(D)} \) be the matrix formed by omitting the row and column corresponding
to the marked arc, we can define the aMVA, the weight system of vMVA on arrow
diagrams as follows.

**Theorem 4.6.** We define

$$aMVA := \frac{det(\widehat{M(D)})}{t_i},$$

where \( t_i \) is the variable associated with the marked arc. This is the weight system associ-
ated to vMVA.

To prove this theorem we must show that for any \( m \) singular link and its corresponding
arrow diagram, if we compute the vMVA of the link, the terms of order \( m - 1 \) are the
same as the terms we get from aMVA on the arrow diagram. The proof is identical to
the proof of Theorem 4.3.
Chapter 5

Circuit Algebras

We introduce the concept due to Dror Bar-Natan [BN] of a circuit algebra, a mathematical object modeled after electrical circuit diagrams. Just as planar algebras provide a nice setting to work with classical tangles, circuit algebras provide the ‘right’ setting for virtual tangles.

5.1 Circuit Algebras

Definition 5.1. A circuit diagram with $n$ inputs is a diagram on a large disk with $n$ internal disks removed. Arcs are drawn going from points on each boundary circle to other points on other boundary circles (possibly the same circle) to indicate a connection or pairing (see Figure 5.1). A $\star$ marks a point along each of the boundary circles so that by starting at the $\star$ and traveling in the counterclockwise direction we may speak of the first, second, third etc... point along a circle. We are interested in the pairing information and not the actual paths, thus two circuit diagrams are considered equivalent if they connect the same points.
Circuit diagrams get their names from electrical circuit diagrams, which contain the connection information for different electrical components, but don’t tell you how to connect the wires in space. As long as output A connects with input B the path the wire takes is irrelevant.

**Definition 5.2.** A circuit algebra is a collection of spaces $V$ and a set of morphisms $\mathcal{P}$, with one vector space $V_n$ for each natural number and one morphism for every circuit diagram. A circuit diagram with $k$ internal disks each with $n_1, n_2, \ldots, n_k$ arcs attached to their boundary circles, and $m$ arcs connecting to the outer circle defines a map from $V_{n_1} \otimes V_{n_2} \otimes \ldots \otimes V_{n_k}$ to $V_m$. So the circuit diagram in Figure 5.1 defines a map $V_4 \otimes V_2 \otimes V_2 \rightarrow V_6$. We require that the morphisms commute with the obvious circuit diagram composition (see Figure 5.3).
Example 5.3. For any real inner product space $W$ let $V_n = W^\otimes n$, diagrammatically we can think of $V_n$ as a disk with $n$ copies of $W$ along the boundary. We can define a pairing map by applying the inner product to the paired components.

This circuit diagram defines a map $\rho : W^\otimes 3 \otimes W^\otimes 4 \to W^\otimes 3$ by,

\[(w_1 \otimes w_2 \otimes w_3) \otimes (w_4 \otimes w_5 \otimes w_6 \otimes w_7) \mapsto \langle w_4 \otimes w_5 \rangle \langle w_6 \otimes w_2 \rangle (w_3 \otimes w_7 \otimes w_1)\]

Example 5.4. Virtual tangles form a circuit algebra, where the space $V_{2n}$ is the space of tangles with $n$ strands, and the circuit algebra morphisms are given by gluing tangles to the circuit diagram.
5.2 Oriented Circuit Algebras

Continuing our analogy with electrical components, often the connection have orientation, the end of the cord you plug into your ipod is different from the one you plug in to your computer. The pairings in oriented circuit diagrams are oriented, each arc has an ‘in’ end and an ‘out’ end. We will need to add orientations to our spaces as well, so they will have outgoing connections and incoming connections.

Definition 5.5. An oriented circuit algebra is a collection of spaces \( V \) and morphisms \( P \).
The spaces \( V_{n,m} \) are indexed by a pair of natural numbers (with \( n \) outgoing components and \( m \) incoming components), and the morphisms are indexed by oriented circuit diagrams.
When we apply morphisms to elements or compose morphisms the orientations must match. As before we require that the morphisms commute with oriented circuit diagram composition.

We will now diagrammatically represent the space \( V_{n,m} \) by a disk with \( n \) outgoing arrows and \( m \) incoming arrows. When we apply a circuit diagram map, we apply the pairings oriented outward to the outgoing arrows (in order starting at the \( \star \)) and the incoming pairings to the incoming arrows. As before the meaning of incoming arrows and outgoing arrows will depend on the particular circuit algebra (see the example below and the following section).

Example 5.6. Given any finite dimensional vector space \( W \) we let \( V_{n,m} = W^n \otimes (W^*)^m \), and take as pairing maps interior multiplication.

This oriented circuit diagram defines the map,
\[
\rho : (W^*)^\otimes 3 \otimes (W^2 \otimes W^*) \rightarrow W \otimes (W^*)^\otimes 3
\]
by,
\[
(\phi_1 \otimes \phi_2 \otimes \phi_3) \otimes (w_4 \otimes w_5 \otimes \phi_6)
\mapsto \phi_2(w_5)(w_4 \otimes \phi_3 \otimes \phi_6 \otimes \phi_1).\]
Clearly, circuit algebras form a category. In the next chapter we will see our two primary examples of circuit algebras, and give a circuit algebra morphism between them.
Chapter 6

Alexander Half Densities, Tangles and the MVA.

We define a circuit algebra morphism between Alexander half densities (to be defined in Section 6.2) and regular virtual tangles. This morphism, tMV A, is an extension of the MVA to regular virtual tangles. A simple renormalization will give us, tMV A′, an invariant of classical tangles. Using this extensions we will give simple proofs that the MVA and the vMVA are finite type invariants.

6.1 The Circuit Algebra of Tangles.

Virtual tangles form an oriented circuit algebra, where \( V_{n,n} \) is the space of tangles with \( n \) strands.

Figure 6.1: A tangle in \( V_{1,1} \) and a tangle in \( V_{2,2} \).
The maps between these tangle spaces are given by gluing the end points of the tangle to the circuit diagram and consider any crossing in the circuit diagram as a virtual crossing. Note that there are different ways of drawing pairings; however, two virtual tangles produced from the same pairing are equivalent.

\[ T_1 \rightarrow T_2 \]

\[ T_1 \rightarrow T_2 \]

**Notation.** When we need to refer to the set of all incoming arcs on a tangle (or other circuit algebra object) we will refer them as \( X^{in} \). Likewise the set \( X^{out} \) refers to the set of outgoing arcs.

### 6.2 Alexander Half Densities

We now define the target spaces for \( tMVA \) our extension of the MVA to tangles. These target spaces, called Alexander half densities, form a circuit algebra. To begin with we need the following notation.

**Notation.** If \( V \) is a vector space with dimension \( n \) we let \( \Lambda^{top}(V) = \Lambda^n(V) \) and if \( n \) is even we let \( \Lambda^{1/2}(V) = \Lambda^{n/2}(V) \). If \( X \) is a finite set let \( V \) be the vector space of formal linear combinations of elements in \( X \), then \( \Lambda^{top}(X) = \Lambda^n(V) \) and \( \Lambda^{1/2}(X) = \Lambda^{n/2}(V) \).

**Definition 6.1.** An Alexander half density (AHD) with incoming strands \( X^{in} \) and outgoing strands \( X^{out} \) is an element of

\[
AHD(X^{in}, X^{out}) := \Lambda^{top}(X^{out}) \otimes \Lambda^{1/2}(X^{in} \cup X^{out})
\]

We require that \( |X^{out}| = |X^{in}| \).
Chapter 6. Alexander Half Densities, Tangles and the MVA.

The Alexander half densities are indexed by the number of incoming and outgoing stands. In order to make this into a circuit algebra we must define a pairing map. Though we write this definition as if we have two inputs, there is an obvious generalization to arbitrarily many inputs. If we have any two elements \( a_1 \otimes p_1 \in \mathrm{AHD}(X_{in1}^{in}, X_{out1}^{out}) \) and \( a_2 \otimes p_2 \in \mathrm{AHD}(X_{in2}^{in}, X_{out2}^{out}) \) and the arcs which we wish to pair have the same label, we define the pairing as below.

**Definition 6.2 (The Gluing map).** Let \( G = (X_{in1}^{in} \cup X_{in2}^{in}) \cap (X_{out1}^{out} \cup X_{out2}^{out}) \) be the set of paired arcs. The pairing map in \( \mathrm{AHD} \) is given by

\[
i_G(a_1 \wedge a_2) \otimes i_G(p_1 \wedge p_2),
\]

where \( i_G \) denotes interior multiplication. Which gives an element in \( \mathrm{AHD}(X_{in1}^{in} \cup X_{in2}^{in} - G, X_{out1}^{out} \cup X_{out2}^{out} - G) \).

Note that to define the map \( i_G \) we must choose an ordering on \( G \), however the sign ambiguity that comes from choosing that ordering is eliminated as \( i_G \) is applied to two terms.

**Theorem 6.3.** Under this gluing map the Alexander half densities form a circuit algebra.

**Proof.** The pairings in the circuit algebra maps tell which arcs to glue together. We must verify that the gluing map commutes with circuit algebra map composition. Suppose we have two compatible circuit algebra maps \( \rho_1 \) and \( \rho_2 \). Let \( G_1 \) and \( G_2 \) be the corresponding sets of paired arcs. The set \( G_1 \cup G_2 \) will be the set of paired arcs for the composition \( \rho_2 \circ \rho_1 \). The corresponding gluing maps written below are equal due to the definition of interior multiplication as,

\[
i_{G_2}(i_{G_1}(a_1 \wedge a_2)) \otimes i_{G_2}(i_{G_1}(p_1 \wedge p_2)) = i_{G_1 \cup G_2}(a_1 \wedge a_2) \otimes i_{G_1 \cup G_2}(p_1 \wedge p_2).
\]

So the gluing maps do commute with circuit algebra maps, making Alexander half densities into a circuit algebra. \( \square \)
6.3 The Multivariable Alexander Polynomial on Tangles

Let $T$ be a tangle with exiting strands labeled by $X^{\text{out}}$ and the incoming strands labeled by $X^{\text{in}}$. If we compute an Alexander matrix as before, it would have the form shown below. Since it is a non-square matrix we cannot take determinants, so we will define our invariant by taking particular minors along with some extra information.

\[
\begin{pmatrix}
\text{Internal Arcs} & X^{\text{out}} & X^{\text{in}} \\
\end{pmatrix}
\]

**Definition 6.4.** The multivariable Alexander polynomial on a regular tangle, $t\text{MVA}$, is

\[
t\text{MVA}(T) = \prod_k t_k^{\frac{-\mu(k)}{2}} \omega \otimes \left( \sum_{i_1 < \ldots < i_n} M(T)^{i_1 \ldots i_n} x_{i_1} \wedge \ldots \wedge x_{i_n} \right),
\]

where $M(T)^{i_1 \ldots i_n}$ is the matrix made up of the columns indexed by internal arcs as well as the columns $i_1, \ldots, i_n$, and $\omega$ is an element of $\Lambda^{top}(X^{\text{out}})$ which records the order of the elements of $X^{\text{out}}$ used in the rows of $M(T)$.

**Theorem 6.5.** $t\text{MVA}$ is a regular tangle invariant.

*Proof.* To show that $t\text{MVA}$ is a regular tangle invariant we must check that it is unchanged by the Reidemeister 2 and 3 moves. This calculation is carried out in Section 8.2.1. \qed

**Theorem 6.6.** The multivariable Alexander polynomial commutes with the previously defined gluing map.

*Proof.* We will prove this for the simple case of gluing one arc; more complicated gluings follow by gluing one arc at a time. There are two ways that we can glue a single arc.
In case one we are gluing two separate tangles together at one arc (labeled $a$ and $a'$ as in Figure 6.2). If we were to follow the construction for each individual tangle, we would compute the matrices $M(T_1)$ and $M(T_2)$. The matrix corresponding to the glued tangle will have the following form:

$$
\begin{pmatrix}
    a & a' \\
    M(T_1) & 0 \\
    1 & -1 \\
    0 & M(T_2)
\end{pmatrix}
$$

Figure 6.2: Gluing two tangles together at one arc.

If we now compute the MVA invariant by expanding all minors along the row $a$, the only non-zero minors will be those which choose one element from column $a$ or column $a'$ but not both. However these are exactly the minors (with appropriate signs) that we get from the gluing map.

In the second case gluing two ends of an arc together will add a row in the matrix with only a 1 and a $-1$ in the columns corresponding to the arcs to be glued. Once again the gluing map will select the non zero minors of the new matrix (with appropriate signs).

Since the tMVA commutes with the gluing map on Alexander half densities we only need to define it on the generators as shown in Figure 6.3.

The previous definition of tMVA is not invariant under Reidemeister 1, since we could not use the normalization use in Definition 3.6 without breaking the virtual Reidemeister 1 invariance. If we wish to restrict ourselves to classical tangles we may renormalize tMVA to get a map invariant under all of the classical Reidemeister moves.

**Theorem 6.7.** If we define $tMVA'$ by renormalizing $tMVA$ as follows:

$$tMVA' := \prod_k t^{\text{rot}(k)} tMVA,$$

Then $tMVA'$ is invariant under all classical Reidemeister moves.
where \( \text{rot}(k) \) is the rotation number of the \( k^{th} \) component. Then \( \text{tMVA}' \) is an invariant of classical tangles.

**Proof.** We need to show that \( \text{tMVA}' \) is invariant under the Reidemeister moves. The extra normalization is unchanged by R2 and R3, so we need only verify that the additional factor of \( t_{\text{rot}(k)}^2 \) cancels the changes produced by the R1 moves, we show one such calculation in Section 8.2.1.

\[ \text{tMVA}' = \text{vMVA} \]

\[ = \text{vMVA} \]
you can see that \( \sum (x_i - 1)C_i = 0 \) where \( x_i \) is the variable associated with \( C_i \), the \( i \)th column. As a consequence, since the variable associated to the \( b \)th arc and the \( a \)th arc is the same, deleting the \( b \)th column instead of the \( a \)th column will change only the sign of the determinant. So the coefficient of \( b \otimes a \) will be negative the coefficient of \( b \otimes b \).

Both the vMVA and the tMVA for a long knot can be computed from the same Alexander matrix. In fact the minor that you take to compute the vMVA is the same minor you use to compute the coefficient of \( b \otimes b \). Their normalizations differ only in that we divide vMVA by a factor of \( t_a - 1 \).

**Corollary 6.9.** If \( T \) is a 1-tangle and \( b \) is the label of the exiting arc, then:

\[
(t_b - 1)vMVA(T) = i_b \otimes i_b(tMVA(T)),
\]

where \( i_b \) denotes interior multiplication.

**Proof.** This is a direct consequence of Theorem 6.8.

To get back the usual MVA, \( \Delta \), we will have to add a normalization as in tMVA'.

**Theorem 6.10.** If \( T \) is any 1-tangle, and \( \text{rot}(k) \) is the rotation number of the \( k \)th component in the tangle \( T \), then

\[
\frac{\sqrt{t_a}}{t_a - 1} \prod_k t_k^{\text{rot}(k)} \ tMVA \left( \begin{array}{c} a \end{array} \begin{array}{c} T \end{array} \rightarrow \begin{array}{c} b \end{array} \right) = \Delta \left( \begin{array}{c} a \end{array} \begin{array}{c} T \end{array} \rightarrow \begin{array}{c} b \end{array} \right) (b \otimes b - b \otimes a).
\]

**Proof.** If you compute an Alexander matrix for the tangle \( T \) as shown above, by adding one row (as shown below) you can get an Alexander matrix for the link produced by closing the two long ends. Since when we compute the multivariable Alexander polynomial for a link we can choose which row and column to delete before taking the determinant, we can compute the MVA by ‘deleting’ that missing row and the column corresponding to the incoming arc \( a \). That minor is the one used to compute the coefficient of \( b \otimes b \).
For the same reasons as in the previous proof the coefficient of $a \otimes b$ will be equal to negative the coefficient of $b \otimes b$.

We still need to account for the different normalizations, since we have decided to delete the $a$th row and column, for $\Delta$ the normalization factor would be $\frac{w_a}{t_a - 1} \prod_k t_k^{\frac{\text{rot}(k) - \mu(k)}{2}}$. However $\omega_a = 1$ since the point to the right of where the $a$th arc starts is already in the unbounded region of the plane. The normalization factor for $tMVA$ would be $\prod_k t_k^{-\frac{\mu(k)}{2}}$, but the rotation number for the strand we are closing will be one less that its rotation number in the closed knot. Hence the difference is $\sqrt{t_a} \frac{1}{t_a - 1} \prod_k t_k^{\frac{\text{rot}(k)}{2}}$, as required.

**Corollary 6.11.** If $T$ is a classical 1-tangle, $a$ is the label of the incoming arc and $b$ is the label of the outgoing arc, and $T_1$ is the link formed by closing the long arc, then:

$$MVA(T_1) = \frac{\sqrt{t_a}}{(t_a - 1)} \prod_k t_k^{\frac{\text{rot}(k)}{2}} i_b \otimes i_b(tMVA(T)),$$

where $i_b$ is interior multiplication.

**The connect sum operation**

It is widely known that the Alexander polynomial is multiplicative under the connect sum operation on knots. The previous lemma provides a simple proof.
Theorem 6.12. Let $\hat{\Delta}$ be the usual single variable Alexander polynomial. Given two knots (links) $K_1$ and $K_2$,

$$\hat{\Delta}(K_1 \# K_2) = \hat{\Delta}(K_1) \hat{\Delta}(K_2).$$

Proof. Let $T_1$ and $T_2$ be two tangles that have all of their arcs labeled by the same variable $x$. Recall that our normalization of the MVA differs from the usual definition of the Alexander polynomial for knots by a factor of $\sqrt{x}(1-x)$. In this case we may rewrite the conclusion of Lemma 6.10 as follows:

$$t\text{MVA} \left( \begin{array}{c}
\begin{array}{c}
K_1
\end{array}
\end{array} \right) = \hat{\Delta} \left( \begin{array}{c}
\begin{array}{c}
T
\end{array}
\end{array} \right) b \otimes (b - a).$$

Let us label the arcs we will connect as below, so that we will connect the two knots by gluing $a$ to $a'$ and $b$ to $b'$.

Then using Lemma 6.10 we knot the following:

$$t\text{MVA} \left( \begin{array}{c}
\begin{array}{c}
K_1
\end{array} \end{array} \right) = \hat{\Delta}(K_1)(a \otimes (ab)), \quad t\text{MVA} \left( \begin{array}{c}
\begin{array}{c}
K_2
\end{array} \end{array} \right) = \hat{\Delta}(K_2)(b' \otimes (b' - a')).$$

So by the gluing map we find:

$$t\text{MVA} \left( \begin{array}{c}
\begin{array}{c}
K_1
\end{array} \end{array} \right) = i_a \otimes i_a[\hat{\Delta}(K_1)(a(\otimes a - b)) \wedge \hat{\Delta}(K_2)(b' \otimes (b' - a'))]$$

$$= \hat{\Delta}(K_1)\hat{\Delta}(K_2)[b' \otimes (b' - b)].$$

But we also know from Lemma 6.10 that:

$$t\text{MVA} \left( \begin{array}{c}
\begin{array}{c}
K_1 \# K_2
\end{array} \end{array} \right) = \hat{\Delta}(K_1 \# K_2)[b' \otimes (b' - b)]$$
so \( \hat{\Delta}(K_1 \# K_2) = \hat{\Delta}(K_1)\hat{\Delta}(K_2) \) as needed.

### 6.4.1 An Alternate Proof that the Multivariable Alexander Polynomial is a Finite Type Invariant.

In Chapter 4 by following the work in [Mur99] and [Dyn97] we showed that under a substitution the MVA is a finite type invariant. The definition of tMVA gives an easy proof of this fact. When we compute the difference between the contributions from a positive crossing and a negative crossing we get the following (for simplicity we have omitted the common first term of \( c \wedge b \)):

\[
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
 c \quad y \quad a \\
d \quad x
\end{array}
\end{array}
\end{array}
\quad - \quad
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
 c \quad y \\
a \quad d \quad x
\end{array}
\end{array}
\end{array}
\end{align*}
\]

\[\mapsto (\sqrt{y} - \frac{1}{\sqrt{x}})a \wedge b + (\sqrt{x} - \sqrt{y})a \wedge c + \left( \frac{1}{\sqrt{x}} - \frac{y}{\sqrt{x}} \right)a \wedge d + (\sqrt{x} - \frac{1}{\sqrt{y}})b \wedge c \wedge d.
\]

Under the substitution \( x = e^{t_1} = 1 + t_1 + ..., \) and \( y = e^{t_2} = 1 + t_2 + ...., \) ignoring higher order terms, we get:

\[(c \wedge b) \otimes (\frac{1}{2}(t_1 + t_2) a \wedge b + \frac{1}{2}(t_1 - t_2) a \wedge c - t_2 a \wedge d + t_1 b \wedge c + \frac{1}{2}(t_1 - t_2) b \wedge d + \frac{1}{2}(t_1 + t_2) c \wedge d).
\]

Therefore since tMVA is multiplicative, if there are \( m \) singular points tMVA will be of degree at least \( m \).

Just as we extended the MVA to tangles we may extend cMVA to parts of chord diagrams. Where by parts of chord diagrams we mean chords drawn on skeleton components made up of long arcs and closed loops. As before we will draw the skeleton components with solid lines and the chords with dotted lines. These diagrams are also taken up to the 4T relation in Figure 2.10. Parts of chord diagrams form a circuit algebra by gluing together the skeleton components.

**Definition 6.13.** We may define, \( c\text{MVA}' \), the MVA weight system of part of a chord diagram by defining it on the generators as in Figure 6.4.
\[ c \wedge b \left( \frac{1}{2} (t_1 + t_2) a \wedge b + \frac{1}{2} (t_1 - t_2) a \wedge c - (t_2) a \wedge d \right) 
+ (t_1) b \wedge c + \frac{1}{2} (t_1 - t_2) b \wedge d + \frac{1}{2} (t_1 + t_2) c \wedge d \right) . \]

Figure 6.4: The value of cMVA' on a chord.

**Lemma 6.14.** If \( D \) is any chord diagram let \( M(D) \) denote the matrix formed by following the rule in Figure 4.2, then:

\[ cMVA(D) = \omega \otimes \left( \sum_{i_1 < \ldots < i_n} M(D)^{i_1, \ldots, i_n} x_{i_1} \wedge \ldots \wedge x_{i_n} \right) , \]

where \( M(D)^{i_1, \ldots, i_n} \) is the matrix made up of the columns indexed by internal arcs as well as the columns \( i_1, \ldots, i_n \), and \( \omega \) is an element of \( \Lambda^{top}(X^{\text{out}}) \) which records the order of the elements of \( X^{\text{out}} \) used in the rows of \( M(D) \).

**Proof.** If we use the rule in Figure 4.2 and the construction in Section 6.3 to compute the value assigned to a chord we get the same value as in 6.4, so the maps agree on the generators. It remains to show that the gluing map will commute with the construction in Lemma 6.14, but this proof is identical to the proof of Theorem 6.6.

As in the case with tMVA we must take care when we wish to compute cMVA of a closed chord diagram from the map cMVA' on parts of closed chord diagrams.

**Theorem 6.15.** If \( C \) is any chord diagram with one long cut skeleton component, then:

\[ cMVA' \left( \begin{array}{c} a \\ C \end{array} \right) = x_b cMVA \left( \begin{array}{c} b \\ C \end{array} \right) (b \otimes (b - a)) , \]

where \( x_b \) is the variable associated to the long arc.

**Proof.** The proof of this relation is similar to the proof for Theorem 6.8. In fact once we see that we could define cMVA' as in Definition 6.13 or as in Lemma 6.14 it is identical to the previous proof.
Corollary 6.16. If $C_0$ is a chord diagram with one cut skeleton component and $b$ is the label of the exiting arc, and $C_1$ is the chord diagram we get by closing the cut component, then:

$$c_{MVA}(C_1) = \frac{1}{t_b} i_b \otimes i_b(c_{MVA'}(C_0)),$$

where $i_b$ is interior multiplication.

Likewise for $v_{MVA}$ we can compute the value associated to a semi-virtual crossing (omitting the common first term of $c \land b$):

$$c \land b \quad \mapsto \quad \left( \frac{1}{\sqrt{x}} - 1 \right) a \land b + \left( \frac{y - 1}{\sqrt{x}} \right) a \land c + \left( 1 - \sqrt{x} \right) a \land d$$

$$+ \left( \frac{1}{\sqrt{x}} - 1 \right) b \land c + \left( \sqrt{x} - 1 \right) c \land d.$$

If we apply the same substitution as before (ignoring higher order terms) we get:

$$(c \land b) \otimes \left( \left( -\frac{1}{2} t_1 \right) a \land b + \left( t_2 \right) a \land c + \left( \frac{1}{2} t_1 \right) a \land d + \left( -\frac{1}{2} t_1 \right) b \land c + \left( \frac{1}{2} t_1 \right) c \land d \right).$$

So we can compute the weight of an arrow diagram by using this value on the arrows and then gluing the appropriate legs together. However once again we must take care when we glue the last legs together.

Definition 6.17. We may define, $a_{MVA}'$, the MVA weight system of part of an arrow diagram by defining it on the generators as in Figure 6.5.

\[
\begin{pmatrix}
 a & c \\
 1 & 2 \\
 d & b
\end{pmatrix} \quad \mapsto \quad c \land b \left( -\frac{1}{2} t_1 a \land b + t_2 a \land c + \frac{1}{2} t_1 a \land d - \frac{1}{2} t_1 b \land c + \frac{1}{2} t_1 c \land d \right).
\]

Figure 6.5: The value of $a_{MVA}'$ on an arrow.

Theorem 6.18. If $A$ is any chord diagram with one long cut skeleton component, then:

Where $x_b$ is the variable associated to the arc we are gluing.

Proof. The proof of this theorem is identical to that of Theorem 6.15 for chord diagrams.
\[
\text{aMVA’ } \left( \begin{array}{c} a \\ \text{A} \\ b \end{array} \right) = x_b \text{aMVA } \left( \begin{array}{c} \text{A} \\ b \\ \text{A} \end{array} \right) (b \otimes (b - a)).
\]

**Corollary 6.19.** If \( A_0 \) is an arrow diagram with one cut skeleton component and \( b \) is the label of the exiting arc, and \( A_1 \) is the arrow diagram we get by closing the cut component, then:

\[
\text{aMVA}(A_1) = \frac{1}{x_b} i_b \otimes j_b(\text{aMVA’}(A_0)),
\]

where \( i_b \) is interior multiplication.

**Proof.** This follows directly from the previous theorem. \( \square \)
Chapter 7

Local Relations.

The Alexander polynomial, the MVA, the vMVA and their weight systems are known to satisfy many relations, the proofs of which are varied and appear in many papers. Using our invariant we can give a uniform computational proof of many of these relations. In this chapter we list many such relations. In chapter 8 we give a program which can be used to verify these relations.

The MVA is known to satisfy non-local relations such as behavior after strand doubling [Tur86], or strand deletion [Con70]. We will only include local relations in this chapter.

In the following three tables we include relations such as the following due to Conway [Con70]:

\[
\begin{align*}
\left\langle \begin{array}{c}
\uparrow \\
\downarrow
\end{array} \right\rangle & - \left\langle \begin{array}{c}
\uparrow \\
\downarrow
\end{array} \right\rangle = (\sqrt{x_1x_2} + \frac{1}{\sqrt{x_1x_2}}) \\
\end{align*}
\]

by which we mean the following:
In Chapter 8 we will verify that these relations hold for tangles, by computing the tMVA. However Theorems 6.8 and 6.10 tell us that if the relation holds locally for tMVA it will also hold for vMVA, in the case of $\Delta$ we must also account for any difference of normalization due to any differences in rotation numbers. So you can replace tMVA in the figure above with your favorite variation ($tMVA'$, vMVA or $\Delta$).

### 7.0.2 About the relations

That the vMVA is an invariant of welded knots (the first relation in the second table), was known by ????? due to relations with the MVA and higher dimensional knot theory. The other relations in the second table are simpler version of the relations known to hold for the classical MVA. It is not known at this time these relations determine vMVA.

Similarly the relations in the fourth table are simpler oriented versions of the relations known to hold for the classical MVA weight system on chord diagrams. H. Murikami showed in [Mur99] that the relations in the third table determine the weight of a chord diagram. It is not known if the same is true of the relations in the fourth table.

### 7.1 Four Tables of Relations
## Relations on Classical Knots and Links.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Equation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Skein Relation</td>
<td>( \frac{1}{\sqrt{x_1 x_2}} = (\sqrt{x_1} - 1 \sqrt{x_2}) )</td>
<td>Also known as Conway’s relation; holds only for strands with the same label [Ale28], [Con70].</td>
</tr>
<tr>
<td>Conway’s Second Identities</td>
<td>( \frac{1}{\sqrt{x_1 x_2}} = (\sqrt{x_1} + 1 \sqrt{x_2}) )</td>
<td>In the single variable version these are a consequence of the skein relation [Con70].</td>
</tr>
<tr>
<td>Conway’s Third Identity</td>
<td>( \frac{1}{\sqrt{x_1 x_2}} = x_1 - \frac{1}{\sqrt{x_1 x_2}} )</td>
<td>This was Conways third identity in the definition of the Conway potential function [Con70].</td>
</tr>
<tr>
<td>J. Murakami’s Fifth Axiom</td>
<td>( \frac{x_2}{x_1} = \sqrt{x_1 x_2} )</td>
<td>This is J. Murakami’s fifth axiom for the Conway potential function [Mur93].</td>
</tr>
<tr>
<td>J. Murakami’s Third Axiom</td>
<td>( (t_1 + \frac{1}{t_1})(t_2 - \frac{1}{t_2}) = (t_1 - \frac{1}{t_1})(t_2 + \frac{1}{t_2}) )</td>
<td>Where ( t_i = \sqrt{x_i} ). This is J. Murakami’s third axiom for the Conway potential function [Mur93].</td>
</tr>
<tr>
<td>Naik-Stanford Doubled Delta Move</td>
<td>( \Delta(K_1 # K_2) = \Delta(K_1) \Delta(K_2) )</td>
<td>Parallel strands must have the same label. If two knots (links) have the same MVA then they differ by finitely many of these moves [NS03].</td>
</tr>
<tr>
<td>Connect Sum</td>
<td>( \Delta(K_1 # K_2) = \Delta(K_1) \Delta(K_2) )</td>
<td>The single variable Alexander polynomial is multiplicative under the connect sum operation, there are coefficients in the multi-variable version [Ale28].</td>
</tr>
</tbody>
</table>
### Relations on Virtual Knots and Links

<table>
<thead>
<tr>
<th>Welded Knots</th>
<th>The MVA is a welded knot invariant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our proof p. 70</td>
<td>+ (\sqrt{x_1} + \frac{1}{\sqrt{x_1}})</td>
</tr>
<tr>
<td>Our proof p. 71</td>
<td>+ (\sqrt{x_1} + \frac{1}{\sqrt{x_1}})</td>
</tr>
<tr>
<td>Our proof p. 71</td>
<td>+ (\sqrt{x_1} + \frac{1}{\sqrt{x_1}})</td>
</tr>
<tr>
<td>Our proof p. 71</td>
<td>+ (\sqrt{x_1} + \frac{1}{\sqrt{x_1}})</td>
</tr>
<tr>
<td>Virtual N-S Doubled Deltas</td>
<td>Parallel strands must have the same label. Two versions of the N-S doubled delta move for virtual knots.</td>
</tr>
<tr>
<td>Our proof p. 71</td>
<td>=</td>
</tr>
<tr>
<td>Our proof p. 71</td>
<td>=</td>
</tr>
</tbody>
</table>
### Relations on the Level of Chord Diagrams

<table>
<thead>
<tr>
<th><strong>Smoothing</strong></th>
<th>![Smoothing Diagram]</th>
<th>Only if both arcs are labeled with $x$ [FOKV97].</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our proof p. 72</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Our proof p. 23</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>No Deep Vertices</strong></td>
<td>![No Deep Vertices Diagram]</td>
<td>This is a consequence of Conway’s second identity for knots and links [Mur99].</td>
</tr>
<tr>
<td><strong>Our proof p. 69</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Blob Cutting</strong></td>
<td>![Blob Cutting Diagram]</td>
<td>This is a direct consequence of the second relation.</td>
</tr>
<tr>
<td><strong>Half Blobs</strong></td>
<td>![Half Blobs Diagram]</td>
<td>On any chord diagram there must be an even number of ‘half blobs’.</td>
</tr>
<tr>
<td><strong>The H Relation</strong></td>
<td>![H Relation Diagram]</td>
<td>For this and the following relation, the simpler single variable version it attained by removing all of the ‘blobs’ [FOKV97].</td>
</tr>
<tr>
<td><strong>Our proof p. 83</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4Y</strong></td>
<td>![4Y Diagram]</td>
<td>$= 0$</td>
</tr>
<tr>
<td><strong>Our proof p. 74</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>H. Murakami’s Second Relation</strong></td>
<td>![H. Murakami’s Second Relation Diagram]</td>
<td>$= 0$</td>
</tr>
<tr>
<td><strong>Our proof p. 75</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>H. Murakami’s Third Relation</strong></td>
<td>![H. Murakami’s Third Relation Diagram]</td>
<td>$= x_i$</td>
</tr>
<tr>
<td><strong>Our proof p. 75</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>H. Murakami’s Fourth Relation</strong></td>
<td>![H. Murakami’s Fourth Relation Diagram]</td>
<td>$= x_i^{-1}$</td>
</tr>
<tr>
<td><strong>Our proof p. 75</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>H. Murakami’s Fifth Relation</strong></td>
<td>![H. Murakami’s Fifth Relation Diagram]</td>
<td>$= 0$</td>
</tr>
<tr>
<td><strong>Our proof p. 75</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These four relations together with the first relation in this table are what H. Murakami used to define the weight system for the MVA recursively [Mur99].
### Relations on Arrow Diagrams

<table>
<thead>
<tr>
<th>Tails</th>
<th><img src="image" alt="Diagram" /> = <img src="image" alt="Diagram" /></th>
<th>A consequence of being a welded knot invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commute</td>
<td><img src="image" alt="Diagram" /> = 0</td>
<td>This one legged ‘blob’ would be 0 if it were a chord diagram.</td>
</tr>
<tr>
<td>Directed No Internal Vertices</td>
<td><img src="image" alt="Diagram" /> = <img src="image" alt="Diagram" /> - <img src="image" alt="Diagram" /></td>
<td>A version of Blob cutting for arrow diagrams</td>
</tr>
<tr>
<td>Single Blobs</td>
<td><img src="image" alt="Diagram" /> = <img src="image" alt="Diagram" /></td>
<td>A directed version of the $H$ relation becomes a $Y$ relation.</td>
</tr>
<tr>
<td>Directed Blob Cutting</td>
<td><img src="image" alt="Diagram" /> = <img src="image" alt="Diagram" /> + <img src="image" alt="Diagram" /></td>
<td></td>
</tr>
<tr>
<td>Our proof p. 76</td>
<td><img src="image" alt="Diagram" /> = <img src="image" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 8

Computation and Proof of Relations

Invariants are of less use if they are not computable, and we cannot claim to understand an invariant if we cannot compute it, so we write programs to do these computations for us. For invariants of knots and links these programs can be tested against known values to verify accuracy, and then can be used to search for patterns. Another way to both use and verify these programs is to use them to prove various relations. This can be both a test for the accuracy of our program, and a useful tool. Indeed the program for $t$MVA on tangles can verify all but one (the connect sum relation which we proved in Section 6.4) of the relations in the previous chapter. This gives one uniform way of proving many relations which arise in different settings.

Below I have included a Mathematica program which will compute the MVA on tangles, chord diagrams, and arrow diagrams. It is available for download as a Mathematica notebook called AlexTangles.nb at www.math.utoronto.ca/jfa. The program is contained in the first cell, which you must evaluate first. The rest of the file contains the computations in this chapter. It operates by first computing the MVA on the basic components which make up the object (crossings or chords ...), it then defines an exterior multiplication, and finally the gluing maps to attach the basic components together. The inputs will have the form $AT[T, n, col]$, where $T$ is the tangle (or chord diagram), $n$ is
the number of strands to be glued, and col lists which arcs are associated to the same variable.

We code links and chord diagrams by labeling all the arcs with a number, starting with internal arcs (arcs that begin and end inside the diagram) then labeling the exiting arcs and finally labeling the incoming arcs. We divide arcs at all of the classical crossings. We can now break our diagram down into simpler parts: either crossings, or chords retaining the labels. Below we list how we encode each of the possible inputs:

\[
\begin{align*}
\text{Pos}[a,b,c,d] & \quad \text{arr} \quad \text{Pos}[a,b,c,d] & \quad \text{arr} \\
\text{Neg}[a,b,c,d] & \quad \text{arr} \\
\text{P}[b,a] & \quad \text{arr} \\
\text{CD}[a,b,c,d] & \quad \text{arr}
\end{align*}
\]

We list all of the basic components of our link separated by commas inside curly brackets. The next entry is $n$ the number of strands we are gluing together. The last entry ‘col’ is a list of which arcs are part of the same tangle component, the program associates the variable $x[i]$ with the arcs listed in the $i^{th}$ part of the list.

\footnote{This is not standard notation, for an explanation see Section \[8.2.4\]}
The output of the program will have the form $cH[\text{list}]$, where the ordered list represents an exterior algebra element and $c$ a coefficient. The program does not list the first component of the Alexander half density as it always is the wedge product of the labels of the exiting arcs taken in increasing order.

**Example 8.1.** In the following example we have properly labeled the diagram, and entered it into the program.

![Diagram](image)

The tangle in terms of generators. The number of legs to be glued. The rule which associates variables to arcs.

When we compute the MVA using the program we get the following,

\[
\text{In[15]} := \text{Simplify[}
\text{AT[\{Neg[9,8,2,1], Pos[1,3,5,4], Pos[2,7,6,3]\}, 3, \{\{1, 8, 5\}, \{9, 2, 6\}, \{7, 3, 4\}\}]]}
\]

\[
\text{Out[15]} = (1/(x[1] \text{Sqrt}[x[2]])
\]

\(-H[\{5, 6, 7\}] - H[\{5, 7, 8\}] + H[\{6, 7, 8\}] +
H[\{5, 6, 9\}] x[1] - H[\{5, 7, 9\}] x[1] + H[\{6, 8, 9\}] x[1] -
H[\{5, 6, 8\}] (-1 + x[3]) - H[\{5, 6, 9\}] x[1] x[3] -
\]

where $-H[\{5, 6, 7\}]$ represents $-(4 \wedge 5 \wedge 6) \otimes (5 \wedge 6 \wedge 7)$. 
8.1 The Program

The program has 3 main parts. The first part of the program \( \text{Int}[K, \text{col}] \) defines the MVA map on the basic components. The last line defines the map \( \text{t}[b, \text{col}] \) from the arcs to the variables associated with that arc.

\[
\text{Int}[K_, \text{col}_] := \\
K / . \{ \\
\text{Pos}[a_, b_, c_, d_] :=
\begin{align*}
& t[a, \text{col}]^{(-1/2)} H[\{a, b\}] + \\
& t[a, \text{col}]^{(-1/2)} (t[b, \text{col}] - 1) H[\{a, c\}] - \\
& t[a, \text{col}]^{(1/2)} H[\{a, d\}] + t[a, \text{col}]^{(-1/2)} H[\{b, c\}] + \\
& t[a, \text{col}]^{(1/2)} H[\{c, d\}],
\end{align*}
\]

\[
\text{Neg}[a_, b_, c_, d_] :=
\begin{align*}
& t[b, \text{col}]^{(1/2)} H[\{a, b\}] - t[b, \text{col}]^{(1/2)} H[\{a, d\}] + \\
& t[b, \text{col}]^{(-1/2)} H[\{b, c\}] + \\
& t[b, \text{col}]^{(-1/2)} (-1 + t[a, \text{col}]) H[\{b, d\}] + \\
& t[b, \text{col}]^{(-1/2)} H[\{c, d\}],
\end{align*}
\]

\[
\text{P}[a_, b_] :=
\begin{align*}
& H[\{a\}] - H[\{b\}],
\end{align*}
\]

\[
\text{CD}[a_, b_, c_, d_] :=
\begin{align*}
& (1/2)(-t[a, \text{col}] - t[b, \text{col}]) H[\{a, b\}] + t[b, \text{col}] H[\{a, c\}] + \\
& (1/2)(-t[a, \text{col}] + t[b, \text{col}]) H[\{a, d\}] + \\
& (1/2)(-t[a, \text{col}] + t[b, \text{col}]) H[\{b, c\}] - \\
& t[a, \text{col}] H[\{b, d\}] + (1/2)*(t[a, \text{col}] + t[b, \text{col}]) H[\{c, d\}],
\end{align*}
\]

\[
\text{Yc}[a_, b_, c_, d_, e_, f_] :=
\begin{align*}
& -t[f, \text{col}] t[c, \text{col}] H[\{a, e, b\}] + \\
& t[a, \text{col}] t[c, \text{col}] H[\{a, e, f\}] - \\
& t[f, \text{col}] t[c, \text{col}] H[\{a, c, b\}] + \\
& t[f, \text{col}] t[a, \text{col}] H[\{a, c, d\}] - \\
& t[f, \text{col}] t[c, \text{col}] H[\{a, b, f\}] - \\
& t[f, \text{col}] t[c, \text{col}] H[\{a, b, d\}] - \\
& t[a, \text{col}] t[c, \text{col}] H[\{e, c, f\}] + \\
& t[f, \text{col}] t[a, \text{col}] H[\{e, c, d\}] + \\
& t[a, \text{col}] t[c, \text{col}] H[\{e, b, f\}] - \\
& t[a, \text{col}] t[c, \text{col}] H[\{e, f, d\}] + \\
& t[f, \text{col}] t[a, \text{col}] H[\{c, b, d\}] + \\
& t[f, \text{col}] t[a, \text{col}] H[\{c, f, d\}],
\end{align*}
\]
\[
\begin{align*}
\text{Arr}[a_-, b_, c_, d_] & :> \\
& -\frac{1}{2} t[a, \text{col}] \, H[\{a, b\}] + t[b, \text{col}] \, H[\{a, c\}] - \\
& \frac{1}{2} t[a, \text{col}] \, H[\{a, d\}] - \frac{1}{2} t[a, \text{col}] \, H[\{b, c\}] + \\
& \frac{1}{2} t[a, \text{col}] \, H[\{c, d\}], \\
\end{align*}
\]

\[
\begin{align*}
\text{HBlob}[a_-, b_] & :> \\
& t[a, \text{col}] \, H[\{a\}] - t[a, \text{col}] \, H[\{b\}]
\end{align*}
\]

\[
t[b_-, \text{col}_-] := x[\text{Position}[\text{col}, b][[1, 1]]];
\]

The second part of the program \text{Prod} defines exterior multiplication of a list of elements.

\[
\begin{align*}
\text{Prod}[\{a_\} & ] := a \\
\text{Prod}[\{\text{l}_-\}_\text{List}] & := \\
& \text{Distribute}[1[[1]] \, ** \, \text{Prod}[\\text{Drop}[\text{l}, 1]]] /.
\end{align*}
\]

\[
\begin{align*}
& (c1_. \, * \, H[g\text{List}]) \, ** \, (c2_. \, * \, H[k\text{List}]) \) :>
\end{align*}
\]

\[
\begin{align*}
& c1 \, c2 \, \text{Signature}[\text{Join}[g]] \, H[\text{Sort}[\text{Join}[g, k]]]
\end{align*}
\]

The last part of the program defines the interior multiplication required to glue the first \(n\) legs together, it has two parts: \text{Proj}, which deletes the variables which we are gluing, and \text{Sig}, which calculates the sign.

\[
\begin{align*}
\text{Proj}[\text{l}_-, n_-] & := \\
& \text{L} / . \, H[\text{l}_-\text{List}] /; \text{Take}[\text{l}, n] \, != \, \text{Range}[n] \rightarrow 0 /.
\end{align*}
\]

\[
\begin{align*}
& \text{H}[\text{l}_-\text{List}] := \, H[\text{Drop}[\text{l}, n]]
\end{align*}
\]

\[
\begin{align*}
\text{Sig}[\text{k}_-, n_-] & := \\
& \text{Signature}[
\end{align*}
\]

\[
\text{List} @@ \text{Join} @@ \text{Apply}[
\text{List}, \text{K} /.
\{\text{Pos}[a_-, b_-, c_, d_] \rightarrow \{c, d\},
\text{Neg}[a_-, b_-, c_, d_] \rightarrow \{c, d\},
\text{CD}[a_-, b_-, c_, d_] \rightarrow \{c, d\},
\text{P}[a_-, b_] \rightarrow \{a\},
\text{Yc}[a_-, b_-, c_, d_, e_-, f_] \rightarrow \{b, d, f\},
\text{Arr}[a_-, b_-, c_, d_] \rightarrow \{c, d\},
\text{Yv}[a_-, b_-, c_, d_, e_, f_] \rightarrow \{a, c, e\},
\text{HBlob}[a_-, b_] \rightarrow \{a\}
\}, \{1\}]}
\]
Chapter 8. Computation and Proof of Relations

The final program computes the MVA by mapping the generators to the appropriate elements (Int), multiplying all of those together (Prod), then deleting the glued labels (Proj) with appropriate signs (Sig).

\[
AT[K_, n_, col_] :=
\]
\[
\text{Expand}[\text{Proj}[\text{Sig}[K, n] \cdot \text{Prod}[\text{Int}[K, col]], n] /.
\]
\[
\text{H}[l_List] :> \text{Signature}[1] \text{H}[\text{Sort}[1]]
\]

8.1.1 Computing the MVA of a Link

Example 8.2. To use this program to compute the MVA of a link \(L\), we must first cut one strand to create a long link on which we can compute the tMVA. We will then use Lemma 6.10 to compute \(\Delta(L)\).

The coefficient of the outgoing strand \(\frac{1 - x_1 + x_1^2}{\sqrt{x_2}}(-1 + x_2)\) is a multiple of the MVA of the closed link. tMVA does not include any normalizations coming from the rotation number so if we wish to recover the normalized MVA we must compute that.
ourselves. According to Lemma 6.10, to recover the normalized MVA for the closed link we must multiply by \( \frac{\sqrt{x_i}}{x_i - 1} \prod_{k} \frac{x_{\text{rot}(k)}}{t_k^2} \), where \( x_i \) is the variable associated with the cut component. For this example we find that the MVA of the closed link is \( x_i^{-1}(1 - x_1 + x_1^2) \), which is the same result as in Example 3.7.

### 8.1.2 Verifying Relations

We can use the above program to verify many of the Alexander relations. The following three examples show what we need to keep in mind when entering possible relations into the program.

**Example 8.3. Conway’s Second Identity - Version 1**

When we wish to compare the MVA on different tangles we must be sure to label all of the external arcs the same way on all of the diagrams, and remember to always label the internal arcs first. We do this below to verify a relation due to Conway [Con70]. To avoid having a different number of arcs in different components we artificially break the arcs on the right hand side up in to 3 pieces.

\[
\begin{array}{c}
\( \sqrt{x_1}x_2 + \frac{1}{\sqrt{x_1}x_2} \)
\end{array}
\]

We can now verify the relation using the code below:

```mathematica
In[17] := Simplify[
    AT[{Pos[5, 6, 2, 1], Pos[1, 2, 4, 3]}, 2, {{3, 2, 5}, {4, 1, 6}}] +
    AT[{Neg[1, 2, 4, 3], Neg[5, 6, 2, 1]}, 2, {{3, 2, 5}, {4, 1, 6}}],
    {x[1] > 0, x[2] > 0}] ==
    Simplify[(Sqrt[x[1]] x[2]) + 1/(Sqrt[x[1]] x[2])] * 
    AT[{P[3, 1], P[1, 5], P[4, 2], P[2, 6]}, 2, {{3, 1, 5}, {4, 2, 6}}]]
Out[17] = True
```

You may notice that on the left hand side we included the restriction that \( x_1 > 0 \) and \( x_2 > 0 \), we did this since Mathematica will not equate \( \sqrt{x} \sqrt{x} = x \) unless you specify that \( x > 0 \).
Example 8.4. The $H$ relation. “The $H$ relation” is known to hold on the weight system of the MVA from computations in a previous paper [Arc08].

When we are working with chord diagrams we can only input a trivalent vertex if all of its legs end on the skeleton, so as in the example below we may need to expand some of the trivalent vertices using the STU relation (see Figure 2.11). We then label all of the arcs.

We can then verify the above relation using Mathematica as follows:

```math
In[11]:= 
AT[{CD[1, 7, 2, 3], Yc[9, 5, 8, 4, 6, 1]}, 1, {{2, 1, 6}, {7, 3}, {4, 8}, {5, 9}}] – 
AT[{Yc[9, 5, 8, 4, 1, 2], CD[6, 7, 1, 3]}, 1, {{2, 1, 6}, {7, 3}, {4, 8}, {5, 9}}] == 
AT[{CD[7, 9, 3, 5], P[1, 6], HBlob[2, 1], HBlob[4, 8]}, 
1, {{2, 1, 6}, {7, 3}, {4, 8}, {5, 9}}] + 
AT[{CD[1, 8, 2, 4], P[1, 6], HBlob[3, 7], HBlob[5, 9]}, 
1, {{2, 1, 6}, {7, 3}, {4, 8}, {5, 9}}] – 
AT[{CD[1, 9, 2, 5], P[1, 6], HBlob[3, 7], HBlob[4, 8]}, 
1, {{2, 1, 6}, {7, 3}, {4, 8}, {5, 9}}] – 
AT[{CD[7, 8, 3, 4], P[1, 6], HBlob[5, 9], HBlob[2, 1]}, 
1, {{2, 1, 6}, {7, 3}, {4, 8}, {5, 9}}]

```
8.2 Computational Proofs of the Relations

This section includes the proofs of the relations from Chapter 7. A worksheet showing their verification (file name AlexTangles.nb) is available at www.math.toronto.edu/jfa/.

8.2.1 The Reidemeister Moves

An alternate proof that tMVA is an invariant of tangles is the following verification that it satisfies the Reidemeister moves. Though there are many ways to orient each of the Reidemeister moves, we have included only one option for each move.

**Reidemeister 1.**

Reidemeister 1 does not hold for tMVA which is only invariant under Reidemeister 2 and 3. However if we add an extra normalization factor $\frac{1}{\sqrt{x_1}}$ to compute tMVA' we get Reidemeister 1 invariance.

```
In[12]:= AT[{Pos[4, 2, 1, 3], P[2, 1]}, 2, {{1, 2, 3, 4}}] ===
Expand[Sqrt[x[1]] AT[{P[3, 1], P[1, 2], P[2, 4]}, 2, {{1, 2, 3, 4}}]]
Out[12]= True
```

**Reidemeister 2.**

```
In[13]:= AT[{Pos[1, 2, 4, 3], Neg[5, 6, 2, 1]}, 2, {{5, 3, 2}, {4, 1, 6}}] ===
Expand[AT[{P[3, 1], P[1, 2], P[2, 4]}, 2, {{1, 2, 3, 4}}]]
```

**Reidemeister 3.**

```
In[14]:= AT[{Neg[9, 8, 2, 1], Pos[1, 3, 5, 4], Pos[2, 7, 6, 3]},
3, {{1, 8, 5}, {9, 2, 6}, {7, 3, 4}}] ===
AT[{Pos[8, 7, 1, 3], Neg[2, 1, 6, 5], Pos[9, 3, 2, 4]},
3, {{1, 8, 5}, {9, 2, 6}, {7, 3, 4}}]
Out[14]= True
```
8.2.2 The Relations on Classical Tangles.

The Skein Relation

This only holds if both arcs are associated to the same variable [Ale28, Con70].

\[
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{1} \\
\text{2}
\end{array} \\
\begin{array}{c}
\text{3}
\end{array}
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{4}
\end{array} \\
\begin{array}{c}
\text{4}
\end{array}
\end{array}
\end{align*}
= \left( \sqrt{x} - \frac{1}{\sqrt{x}} \right)_{\text{3}}
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{1}
\end{array} \\
\begin{array}{c}
\text{2}
\end{array}
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{3}
\end{array} \\
\begin{array}{c}
\text{4}
\end{array}
\end{array}
\end{align*}

In[15]:= Simplify[
    AT[{Pos[3, 4, 1, 2]}, 0, {{1, 2, 3, 4}}] -
    AT[{Neg[3, 4, 1, 2]}, 0, {{1, 2, 3, 4}}] -
    (Sqrt[x[1]] - 1/(Sqrt[x[1]]))* AT[{P[2, 3], P[1, 4]}, 0, {{1, 2, 3, 4}}],
    x[1] > 0] === 0


Conway's Second Identity - Version 2

\[
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{1}
\end{array} \\
\begin{array}{c}
\text{2}
\end{array}
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{3}
\end{array} \\
\begin{array}{c}
\text{4}
\end{array}
\end{array}
\end{align*}
= \left( \frac{\sqrt{x_1}}{x_2} + \frac{\sqrt{x_2}}{x_1} \right)_{\text{2}}
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{2}
\end{array} \\
\begin{array}{c}
\text{1}
\end{array}
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{3}
\end{array} \\
\begin{array}{c}
\text{4}
\end{array}
\end{array}
\end{align*}

In[17]:= Simplify[
    AT[{Pos[1, 6, 3, 2], Pos[2, 5, 4, 1]}, 2, {{4, 2, 6}, {5, 1, 3}}] +
    AT[{Neg[1, 6, 3, 2], Neg[2, 5, 4, 1]}, 2, {{4, 2, 6}, {5, 1, 3}}],
    \{x[1] > 0, x[2] > 0\}] ===
    Simplify[(Sqrt[x[1]/x[2]] + Sqrt[x[2]/x[1]])]
    AT[{P[4, 2], P[2, 6], P[3, 1], P[1, 5]}, 2, {{4, 2, 6}, {5, 1, 3}}],
    \{x[1] > 0, x[2] > 0\}]

Out[17]= True

Conway’s Third Identity

\[
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{1}
\end{array} \\
\begin{array}{c}
\text{5}
\end{array}
\end{array} \\
\begin{array}{c}
\begin{array}{c}
\text{2}
\end{array}
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{3}
\end{array} \\
\begin{array}{c}
\text{4}
\end{array}
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{6}
\end{array} \\
\begin{array}{c}
\text{7}
\end{array}
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\begin{array}{c}
\text{8}
\end{array} \\
\begin{array}{c}
\text{9}
\end{array}
\end{array}
\end{align*}

In[21]:= Factor[Simplify[
    AT[{Pos[9, 3, 1, 4], Neg[1, 2, 6, 5], Neg[8, 7, 2, 3]},
    \{x[1] > 0, x[2] > 0\}]]
3, \{9, 1, 6\}, \{4, 3, 7\}, \{8, 2, 5\}] +
\text{AT[}\{\text{Pos[9, 8, 1, 3]}, \text{Neg[1, 7, 6, 2]}, \text{Pos[3, 2, 5, 4]}\},
3, \{9, 1, 6\}, \{4, 2, 7\}, \{8, 3, 5\}\} - \\
\text{AT[}\{\text{Neg[9, 3, 1, 4]}, \text{Pos[8, 7, 2, 3]}, \text{Pos[1, 2, 6, 5]}\},
3, \{9, 1, 6\}, \{4, 3, 7\}, \{8, 2, 5\}\} - \\
\text{AT[}\{\text{Neg[9, 8, 1, 3]}, \text{Pos[1, 7, 6, 2]}, \text{Neg[3, 2, 5, 4]}\},
3, \{9, 1, 6\}, \{4, 2, 7\}, \{8, 3, 5\}\}], \{x[1] > 0, x[2] > 0, \\
x[3] > 0\}\] === 0

Out[21] = True

J. Murakami's Fifth Axiom.

\begin{align*}
\frac{x_1 - 1}{\sqrt{x_1 x_2}} &= \frac{x_2 - 1}{\sqrt{x_2 x_3}} \\
\frac{x_3 - 1}{\sqrt{x_3 x_1}} &= \frac{t_1 t_2}{t_1 + 1} - \frac{t_2 t_3}{t_2 + 1} \\
\frac{t_3 t_1}{t_3 + 1} &= \frac{t_1 t_2}{t_1 + 1} - \frac{t_2 t_3}{t_2 + 1}
\end{align*}

\text{Where} \\
t_1 = \sqrt{x_1} \\
t_2 = \sqrt{x_2} \\
t_3 = \sqrt{x_3}.

In[425] := t[1] = \text{Sqrt[x[1]]}; \\
t[2] = \text{Sqrt[x[2]]}; \\
t[3] = \text{Sqrt[x[3]]};
Simplify
\[
\]
\[
\]
\[
\text{Out[428]} = \text{True}
\]

Naik-Stanford doubled delta move

This relation only holds if the parallel strands are associated to the same variable.

In[18]:=
\[
\text{state} = \text{state}
\]
8.2.3 The Relations on Virtual Knots and Links.

The MVA is an invariant of welded knots.

Virtual Conway’s Second Identities

\[
\left(\sqrt{x} + \frac{1}{\sqrt{x}}\right) = \left(\sqrt{x} + \frac{1}{\sqrt{x}}\right)
\]

In[19]:= 
AT[{Pos[5, 6, 1, 3], Pos[1, 7, 2, 4]}, 1, 
\{\{5, 1, 2\}, \{4, 7\}, \{6, 3\}\}] ===
AT[{Pos[5, 7, 1, 4], Pos[1, 6, 2, 3]}, 1, 
\{\{5, 1, 2\}, \{4, 7\}, \{6, 3\}\}]

Out[19]= True

In[21]:= 
Simplify[
AT[{Pos[3, 4, 2, 1]}, 0, \{\{2, 3\}, \{1, 4\}\}] +
AT[{Neg[4, 3, 1, 2]}, 0, \{\{2, 3\}, \{1, 4\}\}] - (1 + x[1])/Sqrt[x[1]]
AT[{P[2, 3], P[1, 4]}, 0, \{\{2, 3\}, \{1, 4\}\}], \{x[1] > 0, 
x[2] > 0\}] === 0

Simplify[
AT[{Pos[4, 3, 2, 1]}, 0, \{\{2, 4\}, \{3, 1\}\}] +
AT[{Neg[3, 4, 1, 2]}, 0, \{\{2, 4\}, \{3, 1\}\}], \{x[1] > 0, x[2] > 0\}] ===
AT[{P[2, 4], P[1, 3]}, 0, \{\{2, 4\}, \{3, 1\}\}] (1 + x[1])/Sqrt[x[1]]

Out[21]= True

Out[22]= True
Virtual Conway’s Third Relation

\[
\begin{align*}
\begin{array}{c}
  \begin{array}{c}
    2 & 3 \\
    1 & 4 \\
    7 & 6 \\
  \end{array} \\
  + \\
  \begin{array}{c}
    2 & 3 \\
    1 & 4 \\
    7 & 6 \\
  \end{array} \\
  = \\
  \begin{array}{c}
    2 & 3 \\
    1 & 4 \\
    7 & 6 \\
  \end{array} \\
  + \\
  \begin{array}{c}
    2 & 3 \\
    1 & 4 \\
    7 & 6 \\
  \end{array}
\end{array}
\end{align*}
\]

In[10]:=
Simplify[
  AT[{Pos[7, 5, 1, 2], Neg[1, 6, 4, 3]}, 1, {{7, 1, 4}, {2, 5}, {3, 6}}] +
  AT[{Pos[1, 5, 3, 2], Pos[7, 6, 4, 1]}, 1, {{7, 4}, {2, 5}, {3, 1, 6}}] -
  AT[{Pos[6, 5, 1, 2], Pos[7, 1, 4, 3]}, 1, {{7, 4}, {2, 5}, {3, 1, 6}}] -
  AT[{Pos[1, 5, 4, 2], Neg[7, 6, 1, 3]}, 1, {{7, 1, 4}, {2, 5}, {3, 6}}],
  \{x[1] > 0, x[2] > 0, x[3] > 0\} == 0

Out[10]= True

Virtual versions of J. Murakami’s Fifth Axiom.

\[
\begin{align*}
  \begin{array}{c}
    3 \\
    1 \\
    2 \\
    4
  \end{array} \\
  = (\sqrt{x_2} + \frac{1}{\sqrt{x_2}}) x_1
\end{align*}
\]

and

\[
\begin{align*}
  \begin{array}{c}
    3 \\
    1 \\
    2 \\
    4
  \end{array} = 0
\end{align*}
\]

In[53]:=
Simplify[
  AT[{P[2, 1], Pos[4, 2, 3, 1]}, 2, \{\{1, 2\}, \{3, 4\}\}] ==
  (x[2] - 1)/Sqrt[x[2]] AT[{P[3, 1], P[1, 2], P[2, 4]}, 2, \{\{1, 2, 3, 4\}\}]

AT[{P[2, 1], Neg[4, 2, 3, 1]}, 2, \{\{1, 2\}, \{3, 4\}\}] == 0

Out[53]= True

Out[54]= True

Virtual Naik-Stanford doubled delta

This relation only holds if the parallel strands are associated to the same variable.
8.2.4 The Relations on the Level of Chord Diagrams.

The 4T relation

\[
\begin{align*}
\text{In[21]:=} & \quad \text{AT[}\{\text{CD[1, 6, 2, 3]}, \text{CD[5, 7, 1, 4]}\}, 1, \{\text{3, 6}\}, \{\text{2, 1, 5}\}, \{\text{7, 4}\}\} - \\
& \quad \text{AT[}\{\text{CD[1, 7, 2, 4]}, \text{CD[5, 6, 1, 3]}\}, 1, \{\text{3, 6}\}, \{\text{2, 1, 5}\}, \{\text{7, 4}\}\} == \\
& \quad \text{AT[}\{\text{CD[5, 6, 2, 1]}, \text{CD[7, 1, 4, 3]}\}, 1, \{\text{3, 1, 6}\}, \{\text{2, 5}\}, \{\text{7, 4}\}\} - \\
& \quad \text{AT[}\{\text{CD[7, 6, 4, 1]}, \text{CD[5, 1, 2, 3]}\}, 1, \{\text{3, 1, 6}\}, \{\text{2, 5}\}, \{\text{7, 4}\}\}
\end{align*}
\]

\text{Out[21]= True}

Smoothing chord diagrams

This relation only holds if all involved arcs are associated to the same variable.

\[
\begin{align*}
\text{In[22]:=} & \quad \text{AT[}\{\text{CD[3, 4, 1, 2]}\}, 0, \{\text{1, 2, 3, 4}\}\} == \\
& \quad \text{Expand}[x[1]*\text{AT[}\{\text{P[1, 4]}, \text{P[2, 3]}\}, 0, \{\text{1, 2, 3, 4}\}\}]
\end{align*}
\]

\text{Out[22]= True}

A chord diagram consequence of Conway’s second relation.

In[23]:= \(\text{AT[\{CD[1, 2, 3, 4], CD[5, 6, 1, 2]\}, 2, \{\{3, 1, 5\}, \{4, 2, 6\}\}] - \text{AT[\{CD[1, 6, 3, 2], CD[5, 2, 1, 4]\}, 2, \{\{3, 1, 5\}, \{2, 4, 6\}\}] ==} \\)
\(\text{Expand[x[1] x[2] AT[\{P[3, 1], P[1, 5], P[4, 2], P[2, 6]\}, 2, \{\{1, 3, 5\}, \{2, 4, 6\}\}]])} \\)
Out[23]= True

The blob cutting relation.

A ‘blob’ can be resolved as follows using STU and then the previous relation:

\[
\begin{align*}
\begin{array}{c}
\text{blob} \\
\text{STU} \\
\text{relation}
\end{array}
\end{align*}
\]

So the ‘blob’ cutting relation follows from the previous relation. Since it does not matter how the blobs are connected we will allow ourselves to cut them in half. There must always be an even number of these so-called ‘half blobs’, if we wish we may connect them in any way we want.

\[
\begin{align*}
\begin{array}{c}
\text{blob} \\
\text{half blob}
\end{array}
\end{align*}
\]

No deep vertices

A ‘deep’ vertex is a vertex which does not have any legs touching the skeleton. The following two relations show that the MVA must vanish on any diagram with a deep vertex.

\[
\begin{align*}
\begin{array}{c}
\text{blob} \\
\text{STU}
\end{array}
\end{align*}
\]

In[24]:= \(\text{AT[\{HBlob[3, 6], HBlob[2, 1], CD[7, 5, 4, 1]\}, 1, \{\{3, 6\}, \{7, 4\}, \{1, 2, 5\}\}] ==} \\)
\(\text{AT[\{HBlob[3, 6], HBlob[1, 5], CD[7, 1, 4, 2]\}, 1, \{\{3, 6\}, \{7, 4\}, \{1, 2, 5\}\}]}) \\)
Out[24]= True
Chapter 8. Computation and Proof of Relations

\[
\begin{align*}
\text{In}[25]:= & \quad \text{AT}\{\text{CD}[1, 13, 6, 7], \text{CD}[3, 11, 4, 5], \text{CD}[2, 15, 8, 9], \text{Yc}[12, 1, 14, 2, 10, 3]\}, \\
& \quad \quad \quad 3, \{\{12, 1, 6\}, \{13, 7\}, \{14, 2, 8\}, \{15, 9\}, \{10, 3, 4\}, \{11, 5\}\} - \\
& \quad \text{AT}\{\text{CD}[12, 13, 1, 7], \text{CD}[3, 11, 4, 5], \text{CD}[2, 15, 8, 9], \text{Yc}[12, 1, 14, 2, 10, 3]\}, \\
& \quad \quad \quad 3, \{\{12, 1, 6\}, \{13, 7\}, \{14, 2, 8\}, \{15, 9\}, \{10, 3, 4\}, \{11, 5\}\} - \\
& \quad \text{AT}\{\text{CD}[1, 13, 6, 7], \text{CD}[10, 11, 3, 5], \text{CD}[2, 15, 8, 9], \text{Yc}[12, 1, 14, 2, 10, 3]\}, \\
& \quad \quad \quad 3, \{\{12, 1, 6\}, \{13, 7\}, \{14, 2, 8\}, \{15, 9\}, \{10, 3, 4\}, \{11, 5\}\} - \\
& \quad \text{AT}\{\text{CD}[12, 13, 1, 7], \text{CD}[10, 11, 3, 5], \text{CD}[14, 15, 2, 9], \text{Yc}[12, 1, 14, 2, 10, 3]\}, \\
& \quad \quad \quad 3, \{\{12, 1, 6\}, \{13, 7\}, \{14, 2, 8\}, \{15, 9\}, \{10, 3, 4\}, \{11, 5\}\} = 0
\end{align*}
\]

\[
\text{Out}[25]= \text{True}
\]

\[4Y\]

\[
\begin{align*}
\text{In}[27]:= & \quad \text{AT}\{\text{Yc}[1, 2, 3, 4, 5, 6], \text{HBlob}[8, 7]\}, 0, \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\} - \\
& \quad \text{AT}\{\text{Yc}[3, 4, 5, 6, 7, 8], \text{HBlob}[2, 1]\}, 0, \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\} + \\
& \quad \text{AT}\{\text{Yc}[1, 2, 5, 6, 7, 8], \text{HBlob}[4, 3]\}, 0, \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\} - \\
& \quad \text{AT}\{\text{Yc}[1, 2, 3, 4, 7, 8], \text{HBlob}[6, 5]\}, 0, \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\} = 0
\end{align*}
\]

\[
\text{Out}[27]= \text{True}
\]
H. Murakami’s Second relation

\[
4x_2 \left( \begin{array}{ccc}
2 & 3 & 4 \\
5 & 6 & 7 \\
\end{array} \right) + 2(x_1 - x_3) \left( \begin{array}{ccc}
2 & 3 & 4 \\
5 & 6 & 7 \\
\end{array} \right) + (x_3 - x_1)(x_1 x_3 + x_2^2) \left( \begin{array}{ccc}
2 & 3 & 4 \\
5 & 6 & 7 \\
\end{array} \right) = 0
\]

\[
\text{In[28]} := \text{Expand}
\text{AT}[[\text{CD}[5, 7, 2, 1], \text{CD}[6, 1, 3, 4]], 1, \{\{2, 5\}, \{3, 6\}, \{4, 1, 7\}\}] - \\
\text{AT}[[\text{CD}[5, 6, 1, 3], \text{CD}[1, 7, 2, 4]], 1, \{\{2, 1, 5\}, \{3, 6\}, \{4, 7\}\}] + \\
(2(x[1] - x[3])) \text{AT}[[\text{CD}[5, 1, 2, 3], \text{CD}[6, 7, 1, 4]], 1, \{\{2, 5\}, \{3, 1, 6\}, \{4, 7\}\}] + \\
(2(x[3] - x[1])) (x[1] x[3] + x[2] x[2]) \text{AT}[[\text{P}[2, 5], \text{P}[3, 1], \text{P}[1, 6], \text{P}[4, 7]], 1, \{\{2, 5\}, \{3, 1, 6\}, \{4, 7\}\}] == 0
\]

\[
\text{Out[28]} = \text{True}
\]

H. Murakami’s third relation

\[
\text{In[29]} := \text{AT}[[\text{CD}[4, 2, 3, 1], \text{P}[2, 1]], 2, \{\{3, 4\}, \{2, 1\}\}] == \\
\text{Expand}[x[1] \text{AT}[[\text{P}[3, 2], \text{P}[2, 1], \text{P}[1, 4]], 2, \{\{1, 2, 3, 4\}\}] ]
\]

\[
\text{Out[29]} = \text{True}
\]

8.2.5 The Relations on the level of arrow diagrams.

The 6T relation

\[
\text{In[30]} := \text{AT}[[\text{Arr}[7, 6, 1, 3], \text{Arr}[1, 5, 4, 2]], 1, \{\{7, 1, 4\}, \{5, 2\}, \{3, 6\}\}] + \\
\]
Chapter 8. Computation and Proof of Relations

AT[{Arr[7, 6, 4, 1], Arr[1, 5, 3, 2]}, 1, {{7, 4}, {5, 2}, {3, 1, 6}}] +
AT[{Arr[7, 5, 4, 1], Arr[6, 1, 3, 2]}, 1, {{7, 4}, {5, 1, 2}, {3, 6}}] ===
AT[{Arr[7, 5, 1, 2], Arr[6, 5, 3, 1]}, 1, {{7, 4}, {5, 1, 2}, {3, 6}}] +
AT[{Arr[7, 1, 4, 3], Arr[6, 5, 1, 2]}, 1, {{7, 4}, {5, 2}, {3, 1, 6}}] +
AT[{Arr[7, 1, 4, 2], Arr[6, 5, 3, 1]}, 1, {{7, 4}, {5, 1, 2}, {3, 6}}]

Out[30] = True

Tails Commute

In[31]:=
AT[{Arr[2, 4, 1, 5], Arr[1, 6, 3, 7]}, 1, {{1, 2, 3}, {4, 5}, {6, 7}}] ===
AT[{Arr[2, 6, 1, 7], Arr[1, 4, 3, 5]}, 1, {{1, 2, 3}, {4, 5}, {6, 7}}]

Out[31] = True

Directed no deep vertices, also known as “commutators commute”

When we expand all of the trivalent vertices we get the following sum.

In[32]:=
AT[{Arr[9, 10, 8, 1], Arr[13, 1, 7, 2], Arr[12, 2, 6, 3], Arr[11, 3, 5, 4]},
3, {{1, 2, 3, 4, 10}, {9, 8}, {13, 7}, {12, 6}, {11, 5}}] -
AT[{Arr[12, 10, 6, 1], Arr[11, 1, 5, 2], Arr[9, 2, 8, 3], Arr[13, 3, 7, 4]},
3, {{1, 2, 3, 4, 10}, {9, 8}, {13, 7}, {12, 6}, {11, 5}}] -
AT[{Arr[9, 1, 8, 2], Arr[13, 10, 7, 1], Arr[12, 2, 6, 3], Arr[11, 3, 5, 4]},
3, {{1, 2, 3, 4, 10}, {9, 8}, {13, 7}, {12, 6}, {11, 5}}] -
AT[{Arr[9, 10, 8, 1], Arr[13, 1, 7, 2], Arr[12, 3, 6, 4], Arr[11, 2, 5, 3]},
3, {{1, 2, 3, 4, 10}, {9, 8}, {13, 7}, {12, 6}, {11, 5}}] +
AT[{Arr[9, 1, 8, 2], Arr[13, 10, 7, 1], Arr[12, 3, 6, 4], Arr[11, 2, 5, 3]},
3, {{1, 2, 3, 4, 10}, {9, 8}, {13, 7}, {12, 6}, {11, 5}}],
Chapter 8. Computation and Proof of Relations

\[ 3, \{1, 2, 3, 4, 10\}, \{9, 8\}, \{13, 7\}, \{12, 6\}, \{11, 5\} \]
\[ \text{AT}[[\{\text{Arr}[12, 10, 6, 1], \text{Arr}[11, 1, 5, 2], \text{Arr}[9, 3, 8, 4], \text{Arr}[13, 2, 7, 3]\},
\{1, 2, 3, 4, 10\}, \{9, 8\}, \{13, 7\}, \{12, 6\}, \{11, 5\}] + \]
\[ \text{AT}[[\{\text{Arr}[9, 2, 8, 3], \text{Arr}[13, 3, 7, 4], \text{Arr}[12, 1, 6, 2], \text{Arr}[11, 10, 5, 1]\},
\{1, 2, 3, 4, 10\}, \{9, 8\}, \{13, 7\}, \{12, 6\}, \{11, 5\}] - \]
\[ \text{AT}[[\{\text{Arr}[9, 3, 8, 4], \text{Arr}[13, 2, 7, 3], \text{Arr}[12, 1, 6, 2], \text{Arr}[11, 10, 5, 1]\},
\{1, 2, 3, 4, 10\}, \{9, 8\}, \{13, 7\}, \{12, 6\}, \{11, 5\}] == 0 \]

Out[32]== True

Directed blob cutting

We expand all of the trivalent vertices in the relation.

\[ \text{In}[36]:= \]
\[ \text{AT}[[\{\text{Arr}[5, 6, 1, 2], \text{Arr}[2, 1, 4, 3]\}, 2, \{3, 1, 5\}, \{2, 4, 6\}] - \]
\[ \text{AT}[[\{\text{Arr}[6, 1, 2, 3], \text{Arr}[5, 2, 1, 4]\}, 2, \{3, 1, 5\}, \{2, 4, 6\}] - \]
\[ \text{AT}[[\{\text{Arr}[1, 6, 3, 2], \text{Arr}[2, 5, 4, 1]\}, 2, \{3, 1, 5\}, \{2, 4, 6\}] + \]
\[ \text{AT}[[\{\text{Arr}[6, 5, 2, 1], \text{Arr}[1, 2, 3, 4]\}, 2, \{3, 1, 5\}, \{2, 4, 6\}] == \]
\[ \text{AT}[[\{\text{Arr}[1, 5, 3, 1], \text{Arr}[2, 6, 4, 2]\}, 2, \{3, 1, 5\}, \{2, 4, 6\}] - \]
\[ \text{AT}[[\{\text{Arr}[5, 1, 1, 3], \text{Arr}[2, 6, 4, 2]\}, 2, \{3, 1, 5\}, \{2, 4, 6\}] - \]
\[ \text{AT}[[\{\text{Arr}[1, 5, 3, 1], \text{Arr}[6, 2, 2, 4]\}, 2, \{3, 1, 5\}, \{2, 4, 6\}] + \]
\[ \text{AT}[[\{\text{Arr}[6, 2, 2, 4], \text{Arr}[5, 1, 1, 3]\}, 2, \{3, 1, 5\}, \{2, 4, 6\}] \]

Out[36]== True

A Y relation

\[ \text{A Y relation} \]
When we expand all of the trivalent vertices we get the following sum.

\[
\begin{align*}
\text{In[37]} &:= \text{AT}\{\text{Arr[5, 7, 6, 1], Arr[3, 1, 4, 2]}, 1, \{\{7, 1, 2\}, \{5, 6\}, \{4, 3\}\} - \\
&\quad \text{AT}\{\text{Arr[5, 1, 6, 2], Arr[3, 7, 4, 1]}, 1, \{\{7, 1, 2\}, \{5, 6\}, \{4, 3\}\}\} = \text{AT}\{\text{Arr[5, 7, 6, 2], Arr[1, 3, 4, 1]}, 1, \{\{7, 2\}, \{6, 5\}, \{4, 1, 3\}\} - \\
&\quad \text{AT}\{\text{Arr[5, 7, 6, 2], Arr[3, 1, 1, 4]}, 1, \{\{7, 2\}, \{6, 5\}, \{4, 1, 3\}\}\} + \\
&\quad \text{AT}\{\text{Arr[3, 7, 4, 2], Arr[5, 1, 1, 6]}, 1, \{\{7, 2\}, \{5, 6, 1\}, \{4, 3\}\}\} - \\
&\quad \text{AT}\{\text{Arr[3, 7, 4, 2], Arr[1, 5, 6, 1]}, 1, \{\{7, 2\}, \{5, 6, 1\}, \{4, 3\}\}\}
\end{align*}
\]

\text{Out[37]} = \text{True}
Bibliography


[FOKV97] Jose M. Figueroa-O’Farrill, Takashi Kimura, and Arkady Vaintrob, *The universal Vassiliev invariant for the Lie superalgebra gl(1−1)*, Communications in Mathematical Physics **185** (1997), 93.


