INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Welded Braids and the Crossed Module Invariant

Marta Carvalho Afonso Carreteiro

Dissertação para obtenção do Grau de Mestre em

Matemática e Aplicações

Júri

Presidente:	Doutor Rui António Loja Fernandes
Orientador:	Doutor Francisco Miguel Alves Campos Sousa Dionísio
Co-Orientador:	Doutor João Nuno Gonçalves Faria Martins
Vogal:	Gustavo Rui Gonçalves Fernandes de Oliveira Granja

Setembro de 2009

Resumo

Um módulo cruzado automorfo pode ser definido como um triplo: dois grupos, G e E, o último abeliano, e uma acção de G em E por automorfismos. Esta estrutura pode ser usada para definir um invariante de nós virtuais *welded*.

O objectivo deste trabalho era calcular, recorrendo ao $Mathematica$, o invariante de módulo cruzado de uma extensa classe de nós virtuais *welded* obtidos a partir de tranças *welded* com três fios e até dez cruzamentos, usando módulos cruzados com $G = GL(\mathbb{Z}_p, n)$ e $E = \mathbb{Z}_p^n$. Primeiro definimos o grupo das tranças *welded* e, com base no trabalho já feito por J. F. Martins e L. H. Kauffman, é apresentada uma definição de grafo virtual *welded*. Definimos o invariante de módulo cruzado para nós obtidos a partir de tranças *welded* com m fios usando uma acção do grupo das tranças *welded* com m fios em $(G \times E)^m$ e provamos que é um invariante de nós virtuais *welded* provando a sua invariância em relação aos três tipos de movimentos que podem ser executados numa trança, originando uma trança com o mesmo fecho que a original.

É apresentada uma extensão do invariante de módulo cruzado para grafos virtuais *welded*, a qual é usada para calcular o invariante para um conjunto de exemplos, obtido fechando uma trança sem cruzamentos virtuais de duas maneiras diferentes que têm o *knot group*. Em alguns casos conseguimos distinguir os dois fechos através do invariante de módulo cruzado. Foi calculado um outro exemplo interessante, com *knot group* trivial, apesar de os cálculos feitos não terem distinguido este exemplo do *unknot*.

Palavras-chave: nó virtual *welded*, grupo das tranças *welded*, fecho, invariante de módulo cruzado, grafos virtuais *welded*

Abstract

An automorphic crossed module can be defined as a triple: two groups, G and E, the second one abelian, and an action of G on E by automorphisms. This structure can be used to define an invariant of welded virtual knots.

The aim of this work was to compute, using $Mathematica$, the crossed module invariant of a large class of welded virtual knots arising from welded braids with three strands and up to ten crossings, using crossed modules with $G = GL(\mathbb{Z}_p, n)$ and $E = \mathbb{Z}_p^n$. We first define the welded braid group and, based on the work done by J. F. Martins and L. H. Kauffman, a definition of welded virtual graphs is also presented. We define the crossed module invariant for welded knots coming from welded braids with m strands using an action of the welded braid group on m strands on $(G\times E)^m$ and prove that it yields an invariant for welded virtual knots,by proving its invariance under the three types of moves that can be performed on a braid, which give a braid whose closure is the same as the original one.

We give an extension of the crossed module invariant for welded graphs, and we use it to calculate the inavariant for a set of examples, obtained by closing a braid with no virtual crossings in two different ways that have the same knot group. In some cases we were able to distinguish the two closures using the crossed module invariant. We also computed another interesting example with trivial knot group. However, our calculations could not distinguish this example from the unknot.

Keywords: welded virtual knot, welded braid group, closure, crossed module invariant, welded virtual graphs

Acknowledgements

I would like to express my acknowledgments to everyone who supported me and made this thesis possible, each in its own way and time.

First my advisor, professor Francisco Miguel Dionísio, for the help with the computational component of the thesis.

To my co-advisor, João Martins, I would like to acknowledge the availability and suggestions and for all the meetings that were so important for this work.

I also would like to thank professor Roger Picken, whose guidance along this work has been extraordinary. I want to thank him for all the availability, all the helpful contributions and for making possible the participation of João Martins, whose suggestions were crucial for the development of this thesis.

I must thank Edgar Costa for the help with getting access to the hardware I needed.

To the persons who supported me along these last years: my brother Gonçalo, Ana Domingues and my second family, Tiago Moura, Sara Guerreiro, Joana Leitão, Catarina Leite, Manuel Biscaia, my ERASMUS 'family', Elif Aydin and Cátia Vale. I would like to direct a particular acknowledgment to a special friend, Nuno Freitas, for the support, the words of encouragement and for making everything seem so much easier and fun.

Finally, I thank my mother, who was the person who taught me most so far.

List of Figures

List of Tables

To my mother

Contents

Introduction

Virtual knots were defined for the first time by L. H. Kauffman in 1999, in [8], and can be seen as an extension of classical knots by adding an extra type of crossing - the virtual crossings. In the same work, were also defined welded virtual knots. These knots differ from the virtual knots by the permission of an extra move, and give rise to a similar, but different, theory. Similarly to the classical braid theory already developed and its connections to knots, R. Fenn, R. Rimányi and C. Rourke defined (in [4]) the welded virtual braid group and, in 2000, S. Kamada proved, in [6], the extension of the most important result connecting classical braids and knots - the Markov Theorem - to the welded braid theory; he presented necessary and sufficient conditions for the closure of two different welded braids to give the same welded virtual knot.

The content of this thesis was motivated by the work done by J. F. Martins and L. H. Kauffman in [10]. Our aim was to use *Mathematica* to compute the crossed module invariant of welded virtual knots, defined in their work, using a braid presentation of the knots. This invariant is defined using a finite automorphic crossed module $\mathcal{G} = (G, E, \triangleright)$, that is nothing more than two finite groups G and E (the latter abelian) and a left action $\triangleright$ of G on E by automorphisms, and it was constructed using the crossed module invariant for knotted surfaces already defined by J. Martins in [12].

A knotted surface is an embedding of a closed surface into 4-dimensional space; as particular cases, we have the torus-link, when the embedding is of a disjoint union of tori $S^1 \times S^1$, and a sphere link when the embedding is of a disjoint union of spheres S^2. The relation between knotted surfaces and knots exists due to a map T that sends a knot diagram K to $T(K)$ - the tube of K - which is a torus-link. This map was first defined for classical knots and was extended to virtual knots by S. Satoh, in [13]. Since it was proved that two welded virtual knot diagrams representing the same welded virtual knot have tubes that are isotopic knotted surfaces and that any torus-link is the tube of some welded virtual knot diagram, it seems natural to define a welded virtual knot invariant based on the invariant of knotted surfaces already defined. Thus, the crossed module invariant for welded virtual knots, $\mathcal{H}_{\mathcal{G}}$, presented in this work, is a computation of the crossed module invariant of knotted surfaces, $I_{\mathcal{G}}$, only from the diagram of a welded virtual knot associated to the surface, that is, if we have a knotted surface Σ and a welded virtual diagram K such that $\Sigma = T(K)$ we have $I_{\mathcal{G}}(\Sigma) = \mathcal{H}_{\mathcal{G}}(K)$. However, the invariant $\mathcal{H}_{\mathcal{G}}$ obtained does not require us to remember its origins as a surface invariant, and thus we have a new welded virtual knot invariant regardless of its applications in 4-dimensional space.

This reasoning can be extended to welded virtual arc diagrams, that are embeddings of disjoint unions of intervals $[0,1]$ and circles into $\mathbb{R}^2$. In this case, the tube of a welded virtual arc diagram consisting only of intervals is a sphere-link and the crossed module invariant can be defined in a similar way.

However, the extension to welded virtual graphs is the one with the most interesting applications. With these generalized equivalence classes of diagrams and consequent extension of the crossed module invariant, all defined in [10], it is possible to generate a large number of examples worthy of study. These examples are constructed from closing braids in a special way, in addition to the already known closure; these two closures have the same knot group but some pairs are distinguished by this invariant. In this work, we used the software *Mathematica* to complement the calculations already done in [10]; we did the calculations for the two closures of all braids with three strands and up to ten crossings using the crossed module $\mathcal{G}(n,p) = (GL(\mathbb{Z}_p, n), \mathbb{Z}_p^n, \triangleright)$, where the left action $\triangleright$ of $GL(\mathbb{Z}_p, n)$ on $\mathbb{Z}_p^n$ is the obvious one and for another special example introduced by R. Fenn, that is a non trivial welded virtual knot with the same knot group as the trivial knot.

The text is structured as follows. In chapter 1, we present the fundamental definitions and results in (classical) knot theory. We also introduce the definitions of welded virtual knot and welded braid group; in section 1.3 we define welded virtual graphs and in the following section we present the definition of the knot group for them. The chapter finishes with the explanation of the two different closures for classical braids. In chapter 2, the definition of crossed module is presented and the crossed module invariant is defined for welded virtual braids and is proved that it is a welded virtual knot invariant; in section 2.3 this invariant is extended to welded virtual graphs. Finally, in chapter 3, we explain the implementation done in *Mathematica* of the invariant and we present some of the results obtained for the two closures of the braids and, in 3.3.1, the example introduced by R. Fenn.

Chapter 1

Welded Virtual Knots, Braids and Graphs

In this chapter we present some definitions, examples and results involving knots and braids that will be needed throughout this work.

1.1 Virtual and Welded Knots

A virtual knot diagram can be seen as a classical knot diagram in which an extra type of crossing is allowed, called the virtual crossings. These crossings can be represented as in figure 1.1.

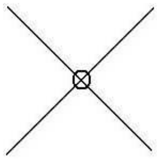

Figure 1.1: Representation of a virtual crossing

Thus we have the definitions:

Definition 1.1.1 *A* ***virtual knot diagram*** *is an immersion of* $S^1 \bigsqcup ... \bigsqcup S^1 \to \mathbb{R}^2$ *injective except at a finite number of double points that can be labeled either with a real or virtual crossing.*

Two virtual knot diagrams are said to be equivalent if one can be transformed into the other by planar isotopy and a finite sequence of real Reidemeister moves together with a version of the Reidemeister moves with virtual crossings and another move involving virtual and classical crossings (see figure 1.2).

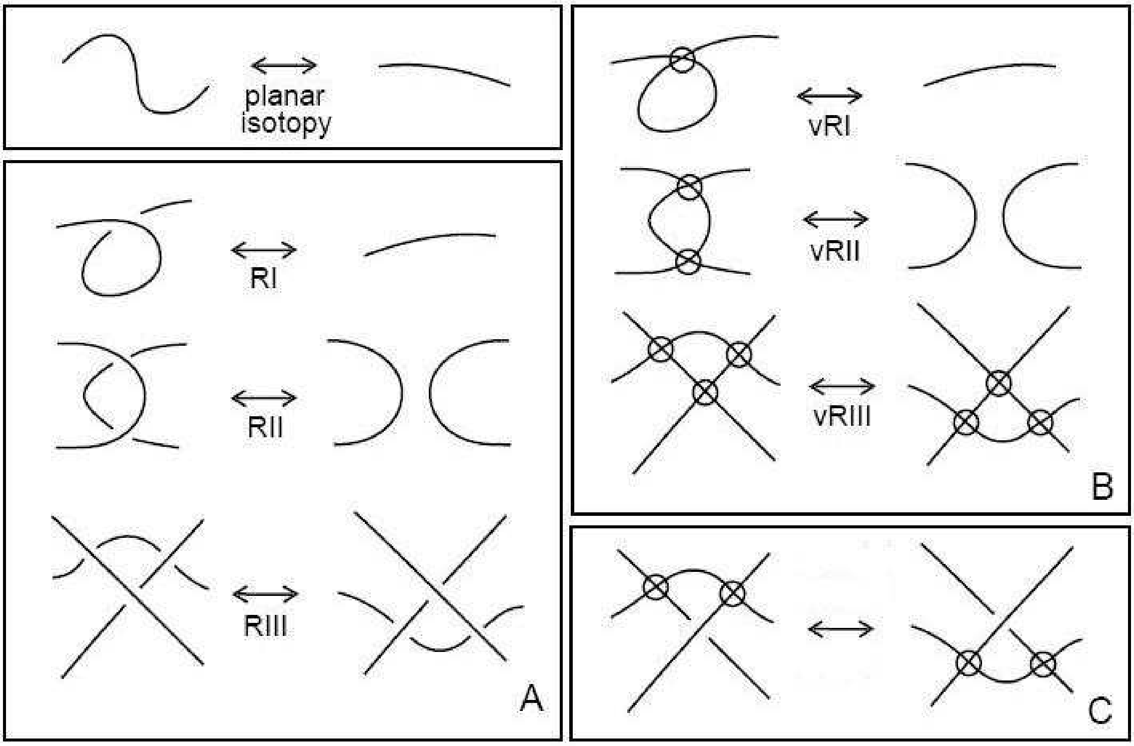

Figure 1.2: Reidemeister moves for virtual knots

Definition 1.1.2 *A* ***virtual knot*** *is an equivalence class of virtual knot diagrams, considering the equivalence relation to be the one described above.*

Remark 1.1.3 *Each circle represented is called a component. When there are two or more components, the equivalence class is called a link. Throughout this text we will do not make the distinction between the two concepts, referring to both as knot.*

Notice there are two moves that do not figure in the list of movements shown above: one with an over arc and a virtual crossing and another one with an under arc and a virtual crossing. These two moves (see figure 1.3) are actually forbidden.

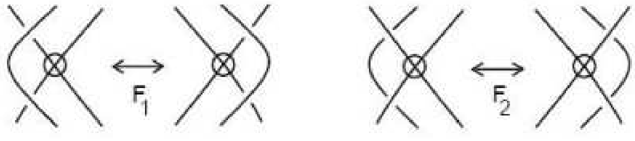

Figure 1.3: Forbidden moves

However, if we allow the move F_1 we obtain what are called ***welded knots***. The virtual knot and the welded virtual knot theories, despite being similar, are distinct. For example, in [13] is presented a knot that is trivial as welded virtual knot but not as virtual (see figure 1.4).

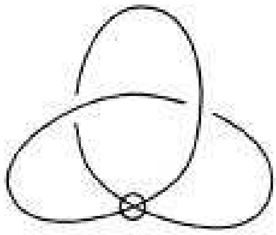

Figure 1.4: A knot which is trivial as a welded virtual knot

As mentioned above, a knot is an equivalence class of knot diagrams. Thus, one can have two different diagrams that represent the same knot. The central problem in knot theory is exactly this, to decide if two different knot diagrams represent, or not, the same knot, which is as yet an unsolved problem, with only partial solutions. The functions called knot invariants are a great help in this task of distinguishing knots apart.

Definition 1.1.4 *A **knot invariant** is a function, defined for all knots, such that its value is the same for equivalent knots.*

It is obvious that to verify if a function is a knot invariant, one just needs to check if it remains the same after the application of each of the moves defining the equivalence relation.

1.2 Welded Braid Group

A very useful presentation of (welded) knots is as a braid. This presentation was used in this work to compute the crossed module invariant.

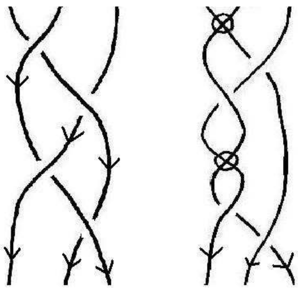

Figure 1.5: Examples of a classical and a welded braid

Definition 1.2.1 *A braid with m strands is a set of non-intersecting curves up to isotopy, in $\mathbb{R}^3$, connecting m distinct points of the line $\{y = 0, z = 0\}$ with $x = 1, ..., m$ to m distinct points of the line $\{y = 0, z = 1\}$ also with $x = 1, ..., m$, such that any plane $z = c, 0 \leq c \leq 1$ intersects each curve exactly once.*

In order to describe a braid, we can encode the crossings as follows: σ_i will represent the crossing in which the ith strand passes under the $(i + 1)th$; σ_i^{-1} is the inverse crossing, when the ith goes over the $(i+1)th$; and finally, the virtual crossing between the ith and the $(i+1)th$ strands will be encoded as τ_i. This encoding is useful to define a group, considering as operation between two braids placing one on the top of the other.

Definition 1.2.2 *The **(Artin) Braid Group**, $\mathcal{B}_m$, on m strands, $m \geq 2$, is the group generated by σ_i, σ_i^{-1}, $i = 1, ..., m - 1$ with the following relations:*

1. $\sigma_i \sigma_i^{-1} = \sigma_i^{-1} \sigma_i = 1$
2. $\sigma_i \sigma_j = \sigma_j \sigma_i$, $|i - j| > 1$
3. $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$

For example, the braid in figure 1.5 on the left is an element of $\mathcal{B}_3$ and is represented as $\sigma_1 \sigma_2^{-1} \sigma_1 \sigma_2^{-1}$.

Similarly, in [4], a group was defined with an extra type of generator (the virtual crossings).

Definition 1.2.3 *The **Welded Braid Group**, $\mathcal{WB}_m$, on m strands, $m \geq 2$, is the group generated by $\sigma_i, \sigma_i^{-1}, \tau_i$, $i = 1, ..., m - 1$ with the same relations as $\mathcal{B}_m$ along with the following:*

4. $\tau_i^2 = 1$
5. $\tau_i \tau_j = \tau_j \tau_i$, $|i - j| > 1$
6. $\tau_i \tau_{i+1} \tau_i = \tau_{i+1} \tau_i \tau_{i+1}$
7. $\sigma_i \tau_j = \tau_j \sigma_i$, $|i - j| > 1$
8. $\sigma_i \tau_{i+1} \tau_i = \tau_{i+1} \tau_i \sigma_{i+1}$
9. $\tau_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \tau_{i+1}$

As an example, look to the braid in figure 1.5 on the right which is an element of $\mathcal{WB}_3$ and it is equal to $\tau_1 \sigma_2 \sigma_1^{-1} \tau_1 \sigma_1 \sigma_2$.

Remark 1.2.4 *Notice that the relation 9. corresponds to the first forbidden move for virtual knots - as presented in figure 1.3, it can be written as* ${\sigma_i}\tau_{i+1}{\sigma_i}^{-1} = {\sigma_{i+1}}^{-1}\tau_i\sigma_{i+1}$ *but multiplying both terms by* σ_{i+1} *on the left and* σ_i *on the right, we obtain 9.. So, if we omit this relation, the group obtained is the Virtual Braid Group on* m *strands,* $\mathcal{VB}_m$*.*

Given a (welded) braid one can close it. Closing a braid is simply to connect corresponding endpoints (i.e., points in $\{y = 0, z = 0\}$ and in $\{y = 0, z = 1\}$ with equal abscissa) using parallel strands.

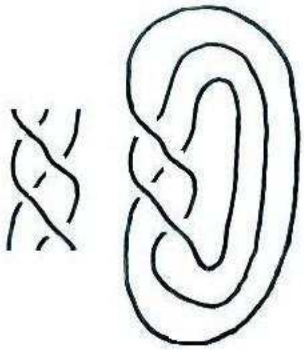

Figure 1.6: A braid and its closure

A result due to Alexander, claims that every link can be represented as a closed braid. This result can be extended to welded knots:

Theorem 1.2.5 *Every welded knot (or link) can be represented by the closure of a welded braid.*

A proof of this result can be found in [6].

Note that different braids can lead to the same welded knot. In classical knot theory, the Markov Theorem gives necessary and sufficient conditions for two braids to give the same knot when closed; this result can also be extended to welded braids (see [6]). Before presenting this result, we need to define some operations that can be done in the welded braid group:

Definition 1.2.6 *Let* $b \in \mathcal{WB}_m$*.*

- b *is said to be obtained by* ***conjugation*** *from* $b_1 \in \mathcal{WB}_m$ *if there exists* $b_2 \in \mathcal{WB}_m$ *such that* $b = b_2^{-1}b_1b_2$*.*
- $\iota_s^t(b)$ *is the element of* $\mathcal{WB}_{m+s+t}$ *obtained by adding* s *trivial strands to the left of* b *and* t *to the right.*

- *A* ***right stabilization*** *of positive, negative or virtual type is a replacement of b by $\iota_0^1(b)\sigma_m$, $\iota_0^1(b)\sigma_m{}^{-1}$ or $\iota_0^1(b)\tau_m$, respectively. Similarly, a* ***left stabilization*** *of positive, negative or virtual type is a replacement of b by $\iota_1^0(b)\sigma_1$, $\iota_1^0(b)\sigma_1{}^{-1}$ or $\iota_1^0(b)\tau_1$, respectively.*

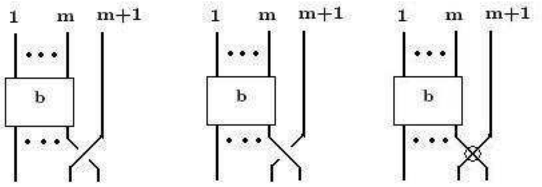

Figure 1.7: Right stabilizations

Theorem 1.2.7 *Two welded braids have equivalent closures as welded links if and only if they are related by a finite sequence of the following moves:*

(WM0) a welded braid move (i.e., a transformation according to the relations defining the welded braid group);

(WM1) a conjugation in the welded braid group;

(WM2) a right stabilization of positive, negative or virtual type and its inverse operation.

Only the right stabilization is considered above; in [6] it is proved that the left stabilizations (all three) are consequences of WM0, WM1 and WM2.

1.3 Welded Virtual Arcs and Welded Virtual Graphs

Definition 1.3.1 *A* ***welded virtual arc diagram*** *is an immersion of unions of intervals $[0,1]$ and circles into the plane $\mathbb{R}^2$, where the 4-valent vertices of the immersion can represent either classical or virtual crossings.*

We define a **welded virtual arc** as an equivalence class of welded virtual arc diagrams, considering two diagrams equivalent if they are related by the moves in figure 1.2 along with the following:

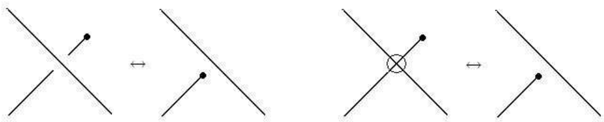

Figure 1.8: Additional moves for welded virtual arcs

Definition 1.3.2 *A **welded virtual graph** is an equivalence class of diagrams, considering the moves for the welded virtual knots and for welded virtual arcs along with the following (whenever a strand is represented with no orientation, the move is legal for any orientation):*

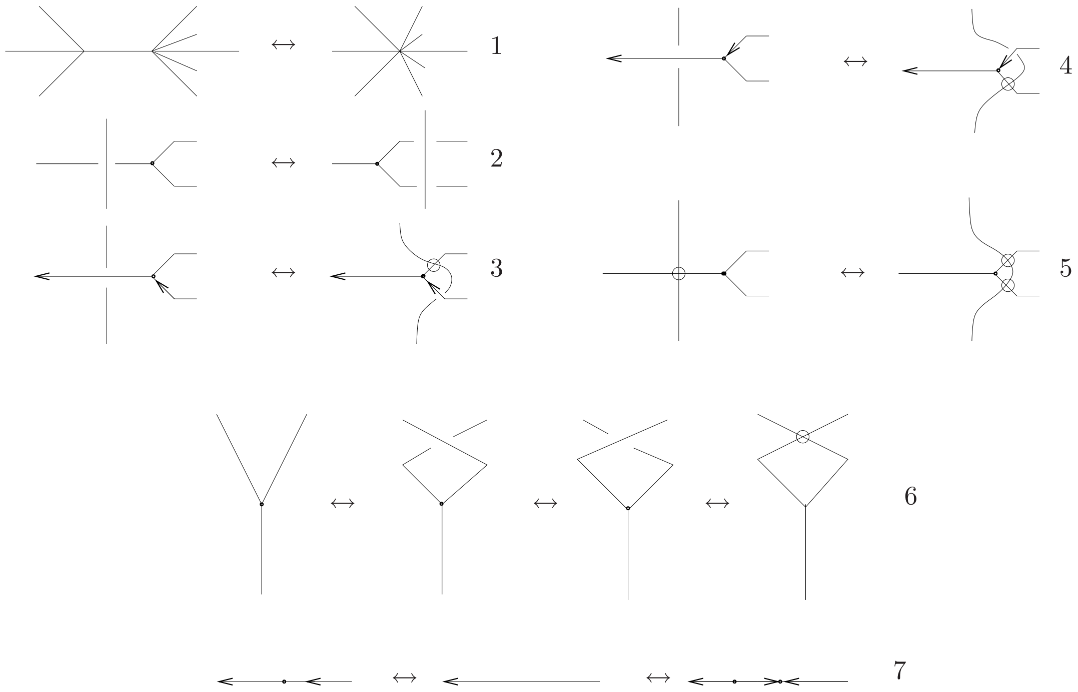

Figure 1.9: Additional moves for welded virtual graphs

Remark 1.3.3 *Notice that the next moves are not allowed:*

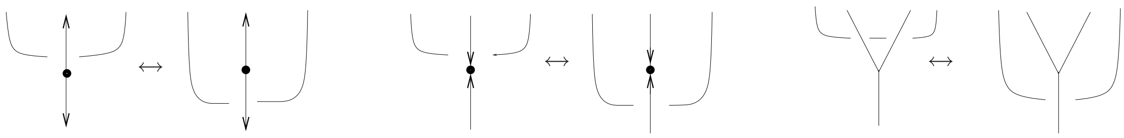

Figure 1.10: Forbidden moves for welded virtual graphs

Given a welded virtual graph K, one can obtain another graph by adding a trivial 1-handle, that consists in choosing a strand of K and doing the next move:

Figure 1.11: Addition of a trivial 1-handle to the graph K

In [10] a way is presented of obtaining a welded virtual knot diagram from a welded virtual graph, using the trivial 1-handle and the moves defining the welded virtual graphs. For this, consider a graph K which is (topologically) the union of circles S^1 and intervals $I = [0, 1]$. Now, if we add a trivial 1-handle to K and then apply a finite sequence of moves, as in figure 1.9, we can obtain another graph K' which is still a union of circles and intervals. However, if K has only one I-component, proceeding as before, we can obtain a graph only with S^1 components which is equivalent to K as a welded virtual graph.

Example 1.3.4 *We can add a trivial 1-handle to the trefoil arc (which if closed as usual, gives the trefoil knot) and then apply a sequence of moves in figures 1.9, 1.8 and 1.2 until we close it. The welded virtual knot obtained is called Shin Satoh's knot.*

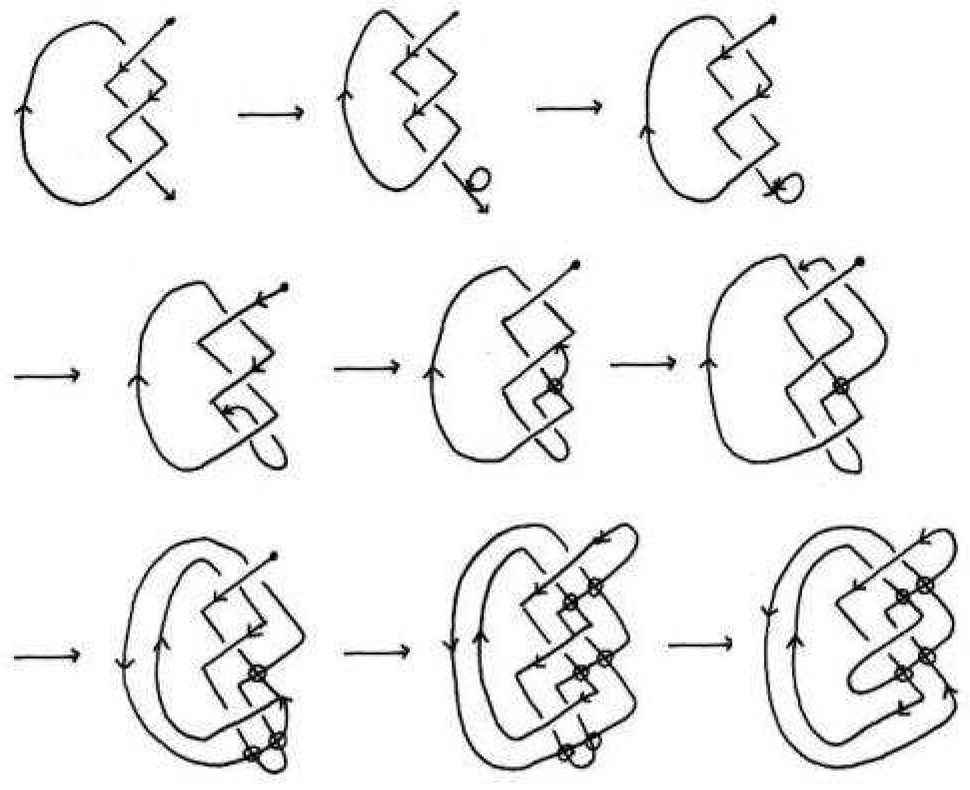

Figure 1.12: Adding a trivial 1-handle to the trefoil arc K

1.4 The Knot Group

The next definition must not be mistaken with the definition of the classical fundamental group of the complement, although throughout the text it will be referred to also as the knot group. Actually, for classical knots, they are the same.

Definition 1.4.1 *The **knot group** of a (welded) virtual knot or a welded virtual arc is the group generated by all the arcs of a diagram of it considering the following relations, the so-called Wirtinger Relations:*

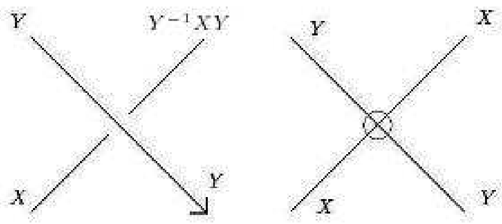

Figure 1.13: Wirtinger relations

Remark 1.4.2 *The concept of arc present in this definition should not be mistaken with the concept of welded virtual arc. Here, an arc is a segment of the strand delimited by consecutive under-crossings or 1-valent vertices*

It can be proved that for arcs with no virtual crossings and with only one component homeomorphic to $[0,1]$,the closure of the arc and the arc itself have the same knot group(see [13]):

Proposition 1.4.3 *Let A be a welded virtual arc with only one component homeomorphic to $[0,1]$ (it can have other components but they must be homeomorphic to S^1) and with no virtual crossings. Consider A sitting in the half space $\{z \geqslant 0\}$ of $\mathbb{R}^3$ and intersecting the plane $\{z = 0\}$ at the end points. Calling K the obvious closure of A, K and A have the same knot group.*

We can also define the knot group for welded virtual graphs, adding the relation in the figure below for the vertices of the graph.

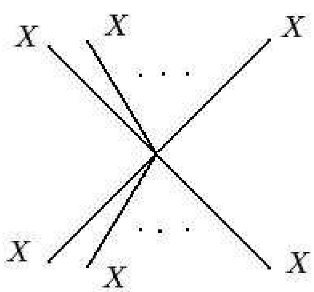

Figure 1.14: Relation of the knot group of a welded virtual graph at a vertex

It can be seen that the knot group is a welded virtual graph invariant.

It is trivial to check that the knot group is also invariant under addition of a trivial 1-handle: suppose we have a welded virtual graph K whose knot group is G and proceed as in figure 1.11; by figure 1.14 we see that this move does not add any generator to G and also does not add also any other relation, since if X is the generator associated with the arc in which the handle is added, the new arc must also be associated with X with no extra conditions.

1.5 Two Different Closures of Classical Braids

An application to classical braids of the previous result leads to a set of very interesting examples that will be used in this work. If we regard an element of $\mathcal{B}_m$ as a welded virtual graph, we can close it in two different ways that will originate two different welded virtual knots with the same knot group. The interesting part is that the crossed module invariant, defined in the next chapter, will distinguish some of these pairs of knots.

Therefore, we will consider the usual closure (see figure 1.15), which will be called the **first trace**, and a second closure, which will be called the **second trace** defined in the following way: consider the usual closure of the braid, but only for the first $m-1$ strands, leaving the last strand untouched. We can close the braid completely by adding a trivial 1-handle to the last strand and then performing the moves in figure 1.9, beginning with the move 1 1.15, thus producing a welded virtual knot; see example below.

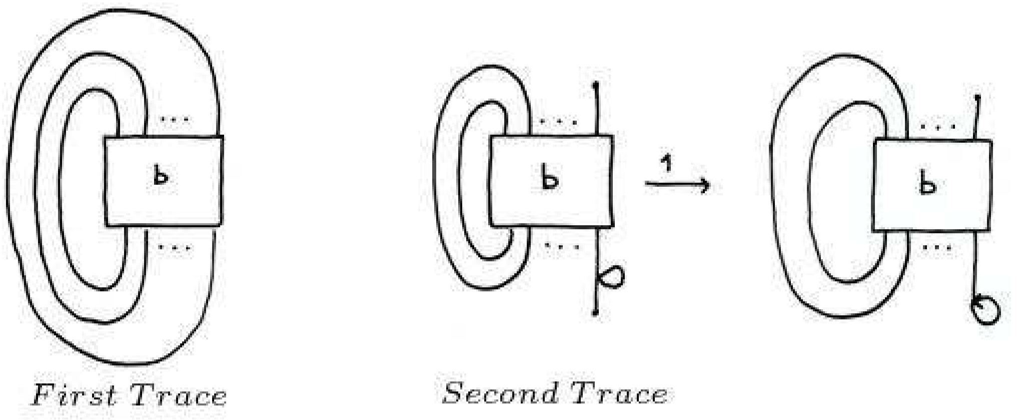

Figure 1.15: The two traces of the a braid b

Example 1.5.1 *The next figure shows the knots that result from the two closures of the same braid - the first trace is the trefoil knot and the second trace is the Shin-Satoh's knot (see figure 1.12).*

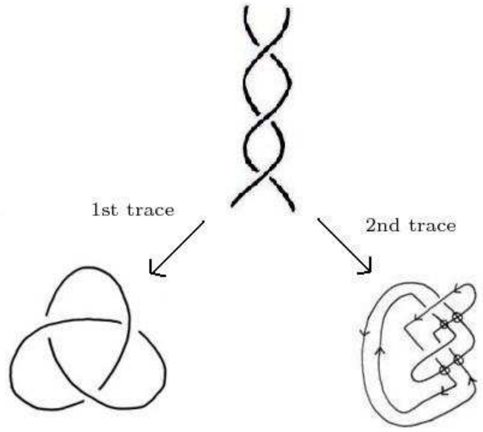

Figure 1.16: The two traces of the same braid: the trefoil knot and the Shin Satoh's knot

We can now observe that by closing a classical braid in these two different ways we obtain two welded virtual knots with the same knot group.

Theorem 1.5.2 *Let $b \in \mathcal{B}_m$, be a classical braid. Let K and K' be the welded virtual knots obtained by the first and second traces of b, respectively. Then, we have that K and K' have the same knot group.*

This result is simple to check just by looking at the construction of the second trace: first we close the first $m-1$ edges by the usual procedure, obtaining a welded virtual arc. Comparing it with the first trace, that is the closure of the arc obtained, we conclude that they have the same knot group (see previous section; prop 1.4.3). Finally, we have the second trace adding a trivial 1-handle to the open component of the arc that does not change the knot group, as we have already seen. Thus, we conclude that the first trace and the second trace of a classical braid have the same knot group (up to isomorphism).

Chapter 2

Crossed Module Invariant

In section 2.1 we define crossed module and give an example. In section 2.2, we present the crossed module invariant for welded virtual knots but starting from a braid presentation of the knots. Finally, in section 2.3 we extend the invariant for welded virtual graphs.

2.1 Crossed Modules

Definition 2.1.1 *Let G and E be groups. A* ***Crossed Module****, $\mathcal{G} = (G, E, \partial, \triangleright)$, is given by a group morphism $\partial : E \to G$ and a left action $\triangleright$ of G on E by automorphisms. The conditions on ∂ and $\triangleright$ are:*

1. $\partial(X \triangleright e) = X\partial(e)X^{-1}$, $\forall X \in G$, $\forall e \in E$
2. $\partial(e) \triangleright f = efe^{-1}$, $\forall e, f \in E$

A crossed module is called ***automorphic*** *when $\partial(e) = 1, \forall e \in E$. In this case, the crossed module is given simply by two groups, G and E, where E is abelian, and a left action of G on E by automorphisms.*

Example 2.1.2 *Let G and E be $GL(\mathbb{Z}_p, n)$ and $\mathbb{Z}_p^n$, respectively, and the action $\triangleright$ simply the product between an element of G and an element of E. This automorphic crossed module will be denoted by $\mathcal{G}(n, p)$.*

2.2 Crossed Module Invariant

Let $\mathcal{G} = (G, E, \triangleright)$ be an automorphic crossed module. We can define an action of the Welded Braid Group, $\mathcal{WB}_m$, on $(G \times E)^m$,$\psi : (G \times E)^m \times \mathcal{WB}_m \to (G \times E)^m$, such that each generator represented in the figure sends a pair in $(G \times E)^2$ to the pair shown below, leaving the other pairs unchanged:

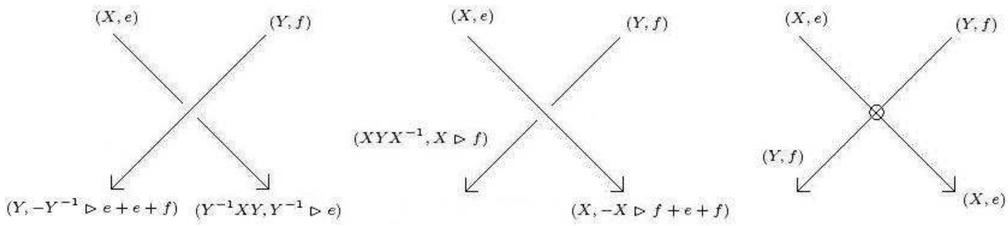

Figure 2.1: Relations at crossings

Theorem 2.2.1 *ψ is a well defined action for the $\mathcal{WB}_m$.*

Proof:

Let $(X, e) = ((X_1, e_1), ..., (X_m, e_m)) \in (G \times E)^m$.

The two properties that define an action are trivially checked: since the identity of $\mathcal{WB}_m$ is the braid with no crossings, it is obvious that $\psi((X, e), 1) = (X, e)$ and it is also clear that we have the equality $\psi((X, e), b_1 b_2) = \psi(\psi((X, e), b_1), b_2)$ since traveling along $b_1 b_2$ from top to bottom is the same as traveling first along b_1 and then along b_2, so $b_1 b_2$ acting on (X, e) is the same as b_1 acting on (X, e) and then b_2 acting on the result of the first action.

Now, we have to prove the invariance under the relations in definition 1.2.3.

Throughout this proof, to simplify the notation, we will only represent the relevant coordinates of the elements in $(G \times E)^m$, that is, if the k-th coordinate is not represented it is of the form (X_k, e_k). Also to simplify the exposition, we will use the following numbered statements repeatedly in the proof. They hold because $\triangleright$ is a left action of G on E.

(1) $e \mapsto X \triangleright e$ is an automorphism

(2) $X \triangleright Y \triangleright e = XY \triangleright e$

(3) $Id \triangleright e = e$, being Id the identity on G

1. $\psi((X, e), \sigma_i \sigma_i^{-1}) = \psi((X, e), \sigma_i^{-1} \sigma_i) = \psi((X, e), 1)$

   According to figure 2.1, we have

$$\psi(((X, e), \sigma_i) = (..., \underbrace{(X_{i+1}, -X_{i+1}^{-1} \triangleright e_i + e_i + e_{i+1})}_{i}, \underbrace{(X_{i+1}^{-1} X_i X_{i+1}, X_{i+1}^{-1} \triangleright e_i)}_{i+1}, ...)$$

   and

$$\psi(\psi((X, e), \sigma_i), \sigma_i^{-1}) = (..., \underbrace{a}_{i}, \underbrace{b}_{i+1}, ...)$$

   where

$$a = (X_{i+1} X_{i+1}^{-1} X_i X_{i+1} X_{i+1}^{-1}, X_{i+1} \triangleright X_{i+1}^{-1} \triangleright e_i) = (X_i, e_i)$$

and

$$
\begin{aligned}
b &= (X_{i+1}, -X_{i+1} \rhd X_{i+1}^{-1} \rhd e_i - X_{i+1}^{-1} \rhd e_i + e_i + e_{i+1} + X_{i+1}^{-1} \rhd e_i) \\
&\overset{(2)}{=} (X_{i+1}, -X_{i+1}X_{i+1}^{-1} \rhd e_i + e_i + e_{i+1}) \\
&\overset{(3)}{=} (X_{i+1}, e_{i+1})
\end{aligned}
$$

So we conclude that

$$\psi((X,e), \sigma_i \sigma_i^{-1}) = (X,e) = \psi((X,e),1)$$

Similarly, we can see that $\psi((X,e), \sigma_i^{-1}\sigma_i) = (X,e)$

2. $\psi((X,e), \sigma_i\sigma_j) = \psi((X,e), \sigma_j\sigma_i)$, $|i-j|>1$

   Here we just need to notice that σ_i acting on (X,e) only changes the ith and (i+1)th coordinates of (X,e) and the action of σ_j of changes the jth and (j+1)th; since $i \neq i+1 \neq j \neq j+1$, it is not relevant if one consider first the action of σ_i or σ_j.

3. $\psi((X,e), \sigma_i\sigma_{i+1}\sigma_i) = \psi((X,e), \sigma_{i+1}\sigma_i\sigma_{i+1})$

$$
\begin{aligned}
&\psi((X,e), \sigma_i\sigma_{i+1}\sigma_i) = \psi(\psi(\psi((X,e),\sigma_i),\sigma_{i+1}),\sigma_i) \\
&= \psi(\psi(((..., \underbrace{(X_{i+1}, -X_{i+1}^{-1} \rhd e_i + e_i + e_{i+1})}_{i}, \underbrace{(X_{i+1}^{-1}X_iX_{i+1}, X_{i+1}^{-1} \rhd e_i)}_{i+1}, ...)), \sigma_{i+1}), \sigma_i) \\
&= (...) = (..., \underbrace{a}_{i}, \underbrace{b}_{i+1}, \underbrace{c}_{i+2}, ...)
\end{aligned}
$$

where

$$
\begin{aligned}
a &= (X_{i+2}, -X_{i+2}^{-1} \rhd [-X_{i+1}^{-1} \rhd e_i + e_i + e_{i+1}] + e_i + e_{i+1} - X_{i+2}^{-1} \rhd X_{i+1}^{-1} \rhd e_i + e_{i+2}) \\
&\overset{(1)}{=} (X_{i+2}, -X_{i+2}^{-1} \rhd [e_i + e_{i+1}] + e_i + e_{i+1} + e_{i+2}) \\
b &= (X_{i+2}^{-1}X_{i+1}X_{i+2}, X_{i+2}^{-1} \rhd [-X_{i+1}^{-1} \rhd e_i + e_i + e_{i+1}]) \\
c &= (X_{i+2}^{-1}X_{i+1}^{-1}X_iX_{i+1}X_{i+2}, X_{i+2}^{-1} \rhd X_{i+1}^{-1} \rhd e_i)
\end{aligned}
$$

On the other hand, we have

$$\psi((X,e), \sigma_{i+1}\sigma_i\sigma_{i+1}) = (...) = (..., \underbrace{a'}_{i}, \underbrace{b'}_{i+1}, \underbrace{c'}_{i+2}, ...)$$

where

$$\begin{aligned}
a' &= (X_{i+2}, -X_{i+2}^{-1} \rhd e_i + e_i - X_{i+2}^{-1} \rhd e_{i+1} + e_{i+1} + e_{i+2}) \\
&\overset{(1)}{=} (X_{i+2}, -X_{i+2}^{-1} \rhd [e_i + e_{i+1}] + e_i + e_{i+1} + e_{i+2}) \\
b' &= (X_{i+2}^{-1} X_{i+1} X_{i+2}, -X_{i+2}^{-1} X_{i+1}^{-1} X_{i+2} \rhd X_{i+2}^{-1} \rhd e_i + X_{i+2}^{-1} \rhd e_i + X_{i+2}^{-1} \rhd e_{i+1}) \\
&\overset{(2)(1)}{=} (X_{i+2}^{-1} X_{i+1} X_{i+2}, X_{i+2}^{-1} \rhd [-X_{i+1}^{-1} \rhd e_i + e_i + e_{i+1}]) \\
c' &= (X_{i+2}^{-1} X_{i+1}^{-1} X_{i+2} X_{i+2}^{-1} X_i X_{i+2} X_{i+2}^{-1} X_{i+1} X_{i+2}, X_{i+2}^{-1} X_{i+1}^{-1} X_{i+2} \rhd X_{i+2}^{-1} \rhd e_i) \\
&\overset{(2)}{=} (X_{i+2}^{-1} X_{i+1}^{-1} X_i X_{i+1} X_{i+2}, X_{i+2}^{-1} X_{i+1}^{-1} \rhd e_i)
\end{aligned}$$

We conclude $\psi((X,e), \sigma_i \sigma_{i+1} \sigma_i) = \psi((X,e), \sigma_{i+1} \sigma_i \sigma_{i+1})$, by observing that $a = a'$, $b = b'$ and $c = c'$.

4. $\psi((X,e), \tau_i^2) = \psi((X,e), 1)$

$$\begin{aligned}
\psi((X,e), \tau_i^2) &= \psi(\psi((X,e), \tau_i), \tau_i) \\
&= \psi((..., \underbrace{(X_{i+1}, e_{i+1})}_{i}, \underbrace{(X_i, e_i)}_{i+1}, ...), \tau_i) \\
&= (..., \underbrace{(X_i, e_i)}_{i}, \underbrace{(X_{i+1}, e_{i+1})}_{i+1}, ...) \\
&= (X,e) = \psi((X,e), 1)
\end{aligned}$$

5. $\psi((X,e), \tau_i \tau_j) = \psi((X,e), \tau_j \tau_i)$, $|i-j| > 1$
Similar to 2.

6. $\psi((X,e), \tau_i \tau_{i+1} \tau_i) = \psi((X,e), \tau_{i+1} \tau_i \tau_{i+1})$ In this case, we have

$$\begin{aligned}
\psi((X,e), \tau_i \tau_{i+1} \tau_i) &= \psi(\psi((..., \underbrace{(X_{i+1}, e_{i+1})}_{i}, \underbrace{(X_i, e_i)}_{i+1}, ...), \tau_{i+1}), \tau_i) \\
&= \psi((..., \underbrace{(X_{i+1}, e_{i+1})}_{i}, \underbrace{(X_{i+2}, e_{i+2})}_{i+1}, \underbrace{(X_i, e_i)}_{i+2}, ...), \tau_i) \\
&= (..., \underbrace{(X_{i+2}, e_{i+2})}_{i}, \underbrace{(X_{i+1}, e_{i+1})}_{i+1}, \underbrace{(X_i, e_i)}_{i+2}, ...)
\end{aligned}$$

and

$$\begin{aligned}
\psi((X,e), \tau_{i+1} \tau_i \tau_{i+1}) &= \psi(\psi((..., \underbrace{(X_{i+2}, e_{i+2})}_{i+1}, \underbrace{(X_{i+1}, e_{i+1})}_{i+2}, ...), \tau_i) \tau_{i+1}) \\
&= (..., \underbrace{(X_{i+2}, e_{i+2})}_{i}, \underbrace{(X_i, e_i)}_{i+1} \underbrace{(X_{i+1}, e_{i+1})}_{i+2}, ...) \\
&= (..., \underbrace{(X_{i+2}, e_{i+2})}_{i}, \underbrace{(X_{i+1}, e_{i+1})}_{i+1}, \underbrace{(X_i, e_i)}_{i+2}, ...)
\end{aligned}$$

Thus, we conclude that $\psi((X,e), \tau_i \tau_{i+1} \tau_i) = \psi((X,e), \tau_{i+1} \tau_i \tau_{i+1})$.

7. $\psi((X,e),\sigma_i\tau_j) = \psi((X,e),\tau_j\sigma_i)$, $|i-j|>1$

   Similar to 2. and 5.

8. $\psi((X,e),\sigma_i\tau_{i+1}\tau_i) = \psi((X,e),\tau_{i+1}\tau_i\sigma_{i+1})$

   We have that

$$
\begin{aligned}
&\psi((X,e),\sigma_i\tau_{i+1}\tau_i) = \psi(\psi(\psi((X,e),\sigma_i),\tau_{i+1}),\tau_i)\\
&= \psi(\psi((...,\underbrace{(X_{i+1},-X_{i+1}^{-1}\triangleright e_i + e_i + e_{i+1})}_{i},\underbrace{(X_{i+1}^{-1}X_iX_{i+1},X_{i+1}^{-1}\triangleright e_i)}_{i+1},...),\tau_{i+1}),\tau_i)\\
&= \psi((...,\underbrace{(X_{i+1},-X_{i+1}^{-1}\triangleright e_i + e_i + e_{i+1})}_{i},\underbrace{(X_{i+2},e_{i+2})}_{i+1},\underbrace{(X_{i+1}^{-1}X_iX_{i+1},X_{i+1}^{-1}\triangleright e_i)}_{i+2},...),\tau_i)\\
&= (...,\underbrace{(X_{i+2},e_{i+2})}_{i},\underbrace{(X_{i+1},-X_{i+1}^{-1}\triangleright e_i + e_i + e_{i+1})}_{i+1},\underbrace{(X_{i+1}^{-1}X_iX_{i+1},X_{i+1}^{-1}\triangleright e_i)}_{i+2},...)
\end{aligned}
$$

   and

$$
\begin{aligned}
&\psi((X,e),\tau_{i+1}\tau_i\sigma_{i+1}) = \psi(\psi(\psi((X,e),\tau_{i+1}),\tau_i),\sigma_{i+1})\\
&= \psi(\psi((...,\underbrace{(X_{i+2},e_{i+2})}_{i+1},\underbrace{(X_{i+1},e_{i+1})}_{i+2},...),\tau_i),\sigma_{i+1})\\
&= \psi((...,\underbrace{(X_{i+2},e_{i+2})}_{i},\underbrace{(X_i,e_i)}_{i+1},\underbrace{(X_{i+1},e_{i+1})}_{i+2},...),\sigma_{i+1})\\
&= (...,\underbrace{(X_{i+2},e_{i+2})}_{i},\underbrace{(X_{i+1},-X_{i+1}^{-1}\triangleright e_i + e_i + e_{i+1})}_{i+1},\underbrace{(X_{i+1}^{-1}X_iX_{i+1}^{-1},X_{i+1}^{-1}\triangleright e_i)}_{i+2},...)
\end{aligned}
$$

9. $\psi((X,e),\tau_i\sigma_{i+1}\sigma_i) = \psi((X,e),\sigma_{i+1}\sigma_i\tau_{i+1})$

$$
\begin{aligned}
&\psi((X,e),\tau_i\sigma_{i+1}\sigma_i) = \psi(\psi(\psi((X,e),\tau_i),\sigma_{i+1}),\sigma_i)\\
&= \psi(\psi((...,\underbrace{(X_{i+1},e_{i+1})}_{i},\underbrace{(X_i,e_i)}_{i+1},...),\sigma_{i+1}),\sigma_i)\\
&= \psi((...,\underbrace{(X_{i+1},e_{i+1})}_{i},\underbrace{(X_{i+2},-X_{i+2}^{-1}\triangleright e_i + e_i + e_{i+2})}_{i+1},\underbrace{(X_{i+2}^{-1}X_iX_{i+2},X_{i+2}^{-1}\triangleright e_i)}_{i+2},...),\sigma_i)\\
&= (...,\underbrace{a}_{i},\underbrace{b}_{i+1},\underbrace{c}_{i+2},...)
\end{aligned}
$$

   where

$$
\begin{aligned}
a &= (X_{i+2},-X_{i+2}^{-1}\triangleright e_{i+1} + e_{i+1} - X_{i+2}^{-1}\triangleright e_i + e_i + e_{i+2})\\
b &= (X_{i+2}^{-1}X_{i+1}X_{i+2},X_{i+2}^{-1}\triangleright e_{i+1})\\
c &= (X_{i+2}^{-1}X_iX_{i+2},X_{i+2}^{-1}\triangleright e_i)
\end{aligned}
$$

On the other hand, we have

$$\psi((X,e),\sigma_{i+1}\sigma_i\tau_{i+1}) = \psi(\psi(\psi((X,e),\sigma_{i+1}),\sigma_i),\tau_{i+1})$$
$$= \psi(\psi((..., \underbrace{(X_{i+2}, -X_{i+2}^{-1} \triangleright e_{i+1} + e_{i+1} + e_{i+2})}_{i+1}, \underbrace{(X_{i+2}^{-1}X_{i+1}X_{i+2}, X_{i+2}^{-1} \triangleright e_{i+1})}_{i+2}, ...),\sigma_i),\tau_{i+1})$$
$$= \psi((..., \underbrace{a'}_{i}, \underbrace{b'}_{i+1}, \underbrace{c'}_{i+2}, ...),\tau_{i+1})$$
$$= (..., \underbrace{a'}_{i}, \underbrace{c'}_{i+1}, \underbrace{b'}_{i+2}, ...)$$

where

$$a' = (X_{i+2}, -X_{i+2}^{-1} \triangleright e_i + e_i - X_{i+2}^{-1} \triangleright e_{i+1} + e_{i+1} + e_{i+2})$$
$$b' = (X_{i+2}^{-1}X_iX_{i+2}, X_{i+2}^{-1} \triangleright e_i)$$
$$c' = (X_{i+2}^{-1}X_{i+1}X_{i+2}, X_{i+2}^{-1} \triangleright e_{i+1})$$

Since $a = a'$, $b = c'$ and $c = b'$, we conclude that $\psi((X,e),\tau_i\sigma_{i+1}\sigma_i) = \psi((X,e),\sigma_{i+1}\sigma_i\tau_{i+1})$.

□

Theorem 2.2.2 *Let $b \in \mathcal{WB}_m$. The number of fixed points of $\phi_b : (G \times E)^m \to (G \times E)^m$, which associates the initial assignment of elements of $G \times E$ to the m strands of b to the final one, according to the action defined above, is a welded virtual knot invariant for the knot that is the first closure of b.*

Proof: according to Theorem 1.2.7, to prove this, we only need to check that the number of fixed points is invariant under the moves (WM0)-(WM2).

The invariance under (WM0) is a direct consequence of the proof of Theorem 2.2.1.

Let $|G| = n_G$ and $|E| = n_E$. Let also $b \in \mathcal{WB}_m$ such that ϕ_b has k fixed points, $((X_i, e_i) = (X_{i1}, e_{i1}), ..., (X_{im}, e_{im})), i = 1, ..., k$.

Let $b' \in \mathcal{WB}_m$ be obtained by conjugation from b - there is a braid $b_1 \in \mathcal{WB}_m$ such that b' is ${b_1}^{-1}bb_1$. Consider an element in $(G \times E)^m$ - $((X,e) = (X_1,e_1), ..., (X_m,e_m))$ such that $\phi_{{b_1}^{-1}}((X,e)) = (X^{(1)}, e^{(1)}) = ((X^{(1)}{}_1, e^{(1)}{}_1), ..., (X^{(m)}{}_m, e^{(m)}{}_m))$. Suppose also that $\phi_b((X^{(1)}, e^{(1)})) = (X^{(2)}, e^{(2)})$ and $\phi_{b_1}((X^{(2)}, e^{(2)})) = (X^{(3)}, e^{(3)})$. In order for (X,e) to be a fixed point of $\phi_{b'}$ one must have $\phi_{b_1}(((X^{(2)}, e^{(2)})) = (X,e)$ but, for this to happen we must have $\phi_{{b_1}^{-1}}((X,e)) = (X^{(2)}, e^{(2)})$ but, in this case $(X^{(2)}, e^{(2)}) = (X^{(1)}, e^{(1)})$, so we conclude that the fixed points of $\phi_{b'}$ correspond to the fixed points of ϕ_b, that is, their number is invariant under conjugation.

Now, let $b' \in \mathcal{WB}_{m+1}$ be obtained by a right negative stabilization of b. To the k fixed points we can "associate" $n_G n_E k$ elements of $(G \times E)^{m+1}$ that are obviously candidates to fixed points of $\phi_{b'}$: for i fixed we have $(X_{i1}, e_{i1}), ..., (X_{im}, e_{im}), (X_{j(m+1)}, e_{j(m+1)})), j =$

$1, ..., n_G n_E$. Applying the relation for the crossing σ_m (see figure 2.2) we conclude that: $X_{im} = X_{j(m+1)}$, so we are reduced to $n_E k$ candidates. Finally, observing that $X_i m \triangleright e_{j(m+1)} = e_{im}$ and that $e \mapsto X \triangleright e$ is an automorphism, we conclude that we have k fixed points.

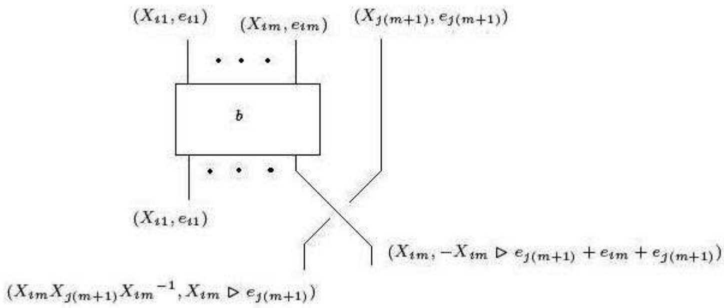

Figure 2.2: $\phi_{b'}((X_{i1}, e_{i1}), ..., (X_{im}, e_{im}), (X_{j(m+1)}, e_{j(m+1)})))$

To check that there are no other fixed points, we need first to observe that the other "candidates" are the elements of $(G \times E)^{m+1}$ obtained by the elements of $(G \times E)^m$ for which ϕ_b only fixes the first $m-1$ components. Lets take one of these elements - $((X_1, e_1), ..., (X_m, e_m))$ whose image by ϕ_b is $((X_1, e_1), ..., (X_{m-1}, e_{m-1}), (X'_m, e'_m))$. After the stabilization, consider the fixed point candidate $((X_1, e_1), ..., (X_{m-1}, e_{m-1}), (X'_m, e'_m), (X_{m+1}, e_{m+1}))$, for $X_{m+1} \in G$ and $e_{m+1} \in E$. Now, we just need to observe the relations:

$X'_m = X_{m+1}$ and $X'_m X_{m+1} {X'_m}^{-1} = X_m$ so we have $X'_m = X_m$

and

$-X'_m \triangleright e_{m+1} + e'_m + e_{m+1} = e_{m+1} \Leftrightarrow -X'_m \triangleright e_{m+1} + e'_m = 0$ and since $X'_m \triangleright e_{m+1} = e_m$ we have $e_m = e'_m$

So we are in the previous case, concluding that, after the stabilization there are exactly k fixed points.

The case of the right positive stabilization of b is similar to the previous one.

Suppose now that b' is obtained from b by a virtual right stabilization. As we have done before, we conclude that the k fixed points of ϕ_b "correspond" to k fixed points of $\phi_{b'}$ - here, since the virtual crossing τ_m only switches the m and $m+1$ components it is immediate to conclude that $X_{im} = X_{j(m+1)}$ and $e_{im} = e_{j(m+1)}$. To see that there are no other fixed points, using the same idea as for the first case, we observe that $(X_m, e_m) = (X_{m+1}, e_{m+1})$ and $(X_{m+1}, e_{m+1}) = (X'_m, e'_m)$, so we are in the case already considered. Again, we have exactly k fixed points for $\phi_{b'}$.

□

2.3 Extension for Welded Virtual Graphs

In [10], the authors difine an extension of the crossed module invariant for welded virtual graphs. With this purpose, a definition is presented for a $\mathcal{G}$-colouring of a welded virtual graph as follows:

Definition 2.3.1 *Let $\mathcal{G} = (G, E, \triangleright)$ be an automorphic crossed module and K an oriented welded virtual graph diagram chosen so that the projection on the second variable is a Morse function in K. A $\mathcal{G}$-colouring of K is an assignment of a pair $(X, e) \in G \times E$ to each arc of K minus its set of critical points and vertices, satisfying the conditions in figure 2.1 along with the ones below:*

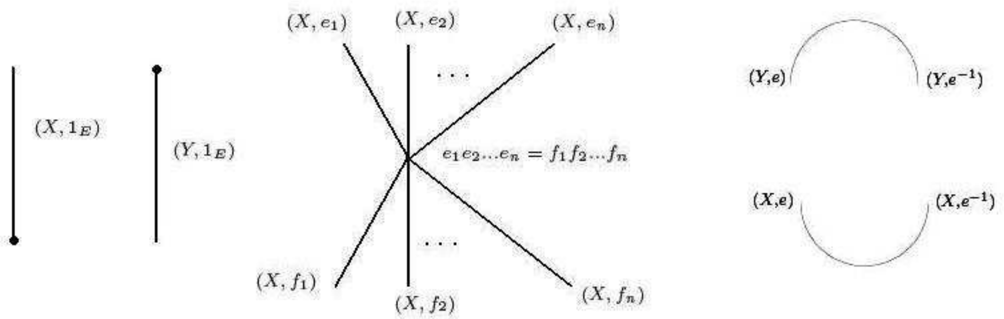

Figure 2.3: $\mathcal{G}$-colourings of welded virtual graphs at endpoints, vertices and at extreme points

Given the notion of $\mathcal{G}$-colouring, we can define the crossed module invariant for welded virtual graphs (see [10]).

Theorem 2.3.2 *Let $\mathcal{G} = (G, E, \triangleright)$ be an automorphic crossed module and K an oriented welded virtual graph diagram.*

$$\mathcal{H}_{\mathcal{G}}(K) = \#\{\mathcal{G} - colourings\ of\ K\} \#E^{\#\{caps\}} \#E^{-\#\{cups\}}$$
$$\#E^{\#\{pointing\ upwards\ 1-valent\ vertices\ of\ K\}}$$
$$\prod_{n-valent\ vert\ v\ of\ K\ n \geq 2} \#E^{1-\{\#edges\ of\ K\ incident\ to\ v\ from\ above\}\}}$$

where the cups and caps are the minimal and maximal points, is a welded virtual graph invariant. Moreover, $\mathcal{H}_{\mathcal{G}}(K) = I_{\mathcal{G}}(T(K))$, where $I_{\mathcal{G}}$ is the crossed module invariant for knotted surfaces defined in [12] and T is the tube map from welded graphs to knotted surfaces defined in [13, 10].

In the next chapter we will show some results of the computation of this invariant. Since we were interested in distinguishing the two traces of a classical braid or in computing the invariant for welded virtual braids, it is useful to observe that for the usual closure of the braids - the first trace (see section 1.5), this invariant is simply the one described in Theorem 2.2.2 and for the second trace of a braid in $\mathcal{B}_m$ it is reduced to:

$$\mathcal{H}_{\mathcal{G}}(K) = \#\{fixed\, points\, such\, that\, e_m = 0\}\#E$$

This holds because, since the two last strands are not connected, the pair (X, e) assigned to the top of the last strand must have $e = 1_E$. The factor $\#E$ appears because, when dealing with braids, there are no n-valent vertices, so the last factor of the invariant is 1;the number of minimal and maximal points is equal and there is one pointing upwards end.

Adding a trivial 1-handle does not change this last expression. After the addition of it, we have a new 3-valent vertex with an edge incident from above, and one more cup but we also have that the $\mathcal{G} - colourings$ are $\mathcal{H}_{\mathcal{G}}(K) = \#\{fixed\, points\, such\, that\, e_m = 0\}\#E$ because to the arc added we can associate an element (X, e) of $G \times E$ with X fixed (it must be the same as the X associated to the arc in which the handle was added) and e arbitrary. Thus we have the same expression for the invariant.

Chapter 3

Computation of the Invariant and Results

The Algorithm developed in $Mathematica$ computes the invariant $\mathcal{H}_{\mathcal{G}}$ mentioned previously with $G = GL(\mathbb{Z}_p, n)$, $E = \mathbb{Z}_p^n$ and the obvious action of G on E, given a welded virtual braid. To obtain the results, the inputs needed are a representation of the braid, n and p.

3.1 The Braid Representation

The braids are represented as lists with two elements, the first being the number of strands and the second another list whose elements encode the crossings. Traveling along the braid from the top to the bottom, the crossings are represented in the list in the same order as they occur in the braid, the classic crossings as integers: the crossing σ_i corresponds to $-i$; the inverse crossing, σ_i^{-1}, corresponds to i; and the virtual crossings are encoded by τi. For example, $\{2, \{-1, -1, -1\}\}$ is the representation of the braid whose closure is the trefoil knot. This encoding might seem not very intuitive. The reason is that *Mathematica* uses another version of our notation (i.e., the positive crossing is the i-th strand passing over the $i+1$-th).

Using the package KnotTheory for *Mathematica*, we can determine a braid representation of any knot listed in the Knot Atlas (see [1]). This representation is called the minimum braid representative - which is unique - of the knot and is minimum in the sense that it first minimizes the number of crossings and then minimizes the number of strands. The algorithm to determine this minimum braid representation can be found in [5].

Remark 3.1.1 *Most of the examples calculated are represented as a classical braid (the virtual crossings appear when we close the braid according to the second trace), but the algorithm is prepared to receive also welded braids - in this case, the closure is done only using the first trace.*

3.2 Computation of the Invariant

As mentioned previously, we want to compute the invariant for pairs of welded virtual knots that arise from the same classical braid by closing it in two different ways, yielding pairs of (possibly distinct) welded knots with the same knot group.

Since the crossed module invariant is a welded virtual graph invariant, we can calculate it for these pairs of welded virtual knots without needing their diagrams. We just need the representation of the braid b with m strands and no virtual crossings and then use the two different expressions for the invariant (see chapter 2):

$$\mathcal{H}_{\mathcal{G}}(1st\ trace\ of\ b) = \#\{fixed\ points\ of\ \phi_b\}$$

$$\mathcal{H}_{\mathcal{G}}(2nd\ trace\ of\ b) = \#\{fixed\ points\ of\ \phi_b\ such\ that\ e_m = 0\}\#E$$

which makes the computation really simple.

Before computing the invariant we needed to generate all the invertible $n \times n$ matrices with entries in $\mathbb{Z}_p$ and all the elements of $\mathbb{Z}_p^n$. Afterwards, we split the calculation into two functions: one to calculate the invariant when the braids are closed according to the first trace and the other when the closure is done according to the second trace. The first function receives as input, as mentioned above, the representation of the braid, the dimension of the matrices and p. The calculation begins with the creation of all pairs (M, v) with $M \in GL(\mathbb{Z}_p, n)$ and $v \in \mathbb{Z}_p^n$. Then, all the tuples of m of these pairs (m being the number of strands that constitute the braid) are checked one by one: for each tuple (a tuple corresponds to a labeling of the strands of the braid) a function is applied that, after a crossing, relabels the strands according to the rules mentioned previously and returns the final labeling, which is compared with the original one. The output obtained is the number of these fixed points. The function used for the second trace is almost identical to this, except for the construction of the tuples - in this case the label for the final strand is a pair with the second component trivial - and in the end the number of fixed points needs to be multiplied by the cardinal of $\mathbb{Z}_p^n$ (see section 2.3).

Remark 3.2.1 *Notice that this algorithm depends on the number of strands: since we do not construct all the m-tuples at the start and then check one by one (this was done at the beginning of this work but, due to limitations of memory of the computers available, was abandoned), we need to introduce a cycle for each strand to check all the tuples. Constructing the tuples at the start, we could give as input the number of strands and create the tuples with that length.*

The calculations were done for the crossed modules $\mathcal{G}(2, 2)$, $\mathcal{G}(2, 3)$ and, in a few cases, $\mathcal{G}(2, 4)$. Since we were working with a very large number of combinations of matrices and vectors and doing comparisons for all these combinations, the computation of the invariant took a lot of time. With the aim of minimizing the time needed and, consequently, to obtain more results, we used the servers of the Mathematics Department of IST. Therefore, we had access to six machines with four processors each. For the calculations with $\mathcal{G}(2,3)$ we used each processor to run the code for one braid (each braid took about one day to finish). But, for $\mathcal{G}(2,4)$ another strategy was used: for each braid, we divided the calculation of the invariant for the first trace in twenty-four parts and of the second trace in four parts and each part ran in one processor. This was done by changing the function that calculates the invariant: we split the cycles into smaller cycles so in each processor few combinations were checked. Using this method, the computation of the invariant, for the two traces, with $\mathcal{G}(2,4)$ took also about one day to complete.

3.3 Examples and Results

3.3.1 Special Example

This example was presented by R. Fenn in [3]. It is a really interesting example for it is a non-trivial welded virtual knot with the same knot group as the trivial knot. This knot, let us call it K, is the closure of the following welded virtual braid that, as seen in section 1.2 is equal to $\tau_1\sigma_2\sigma_1^{-1}\tau_1\sigma_1\sigma_2$ (thus we use as input $\{3,\{\tau 1,-2,1,\tau 1,-1,-2\}\}$).

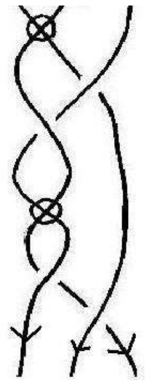

Figure 3.1: A braid presentation of K

Again in [3], there is a brief proof of the non-triviality of this example using the fundamental biquandle.

The crossed module invariant does not distinguish K from the unknot when the crossed module is $\mathcal{G}(2,2)$, $\mathcal{G}(2,3)$ or $\mathcal{G}(2,4)$. The results are shown below:

Table 3.1: K and the unknot

knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	$\mathcal{H}_{\mathcal{G}(2,3)}$	$\mathcal{H}_{\mathcal{G}(2,4)}$
K	24	432	1536
unknot	24	432	1536

The calculation of $\mathcal{H}_{\mathcal{G}(2,5)}(K)$ is being done but it was not finished in time to be presented here.

Since K has the same knot group as the unknot and the map T preserves the knot group, it follows that $T(K)$ has the same fundamental group of the complement as the trivial knotted torus. As mentioned by Fenn in [3], there are some reservations about the proof that this implies that there exists an isotopy between the two corresponding knotted surfaces, but if this is the case then $\mathcal{H}_{\mathcal{G}}$ cannot distinguish K from the trivial knot since $\mathcal{H}_{\mathcal{G}}$ factors through the tube map. Nevertheless this calculation gives a coherence test to the invariant $\mathcal{H}_{\mathcal{G}}$.

3.3.2 Distinction of The Two Traces of The Same Braid

Remark 3.3.1 *In this section, we will use the enumeration for the knots and links used in the Knot Atlas (generally the label for knots is the number of crossings of the knot index by an enumeration). For example,* 3_1 *is the label of the trefoil knot and L6a1 is the first alternating link with six crossings. In the table presented below, we use the label of the knot,* K*, to refer to the first trace of the minimum braid representation of* K *and* $c(K')$ *when referring to the second trace.*

As seen in section 1.5, an interesting set of examples to check are the pairs of welded virtual knots, with the same knot group, obtained by the two closures of the same braid (with no virtual crossings), thus continuing with the work done by J. F. Martins and L. H. Kauffman in [10]. The aim of this work was to compute the invariant for all the braids with three strands and with up to ten crossings, since the closure of these braids gives rise to a set of knots already significant to study and these calculations could be done in the time available for this project.

However, it is not necessary to check all the combinations of ten crossings involving three strands, because some of these give rise to equivalent braids (using the relations in the definition of welded braid group - definition 1.2.3) and also because some, when closed, originate the same knot (when they are related by a sequence of moves presented in theorem 1.2.7). Since we already have seen that this invariant is a welded virtual knot invariant (see theorem 2.2.2) it would be irrelevant to do the calculations for all these braids that are equivalent or that represent the same knot.

Therefore, with the purpose of eliminating these cases, we proceeded as follows. If we start with all the 3-strand braids with up to ten crossings and we close them, since the process of closing does not add any crossings, we get also knots with up to ten crossings. At this point, since these knots are already listed (consult [1]), we are reduced to 250 knots so we will have, at most, this number of braids. Now, using *Mathematica* we determine the minimum braid representative and choose those which are represented by a 3-strand braid. So we are reduced to 57 braids.

Finally, we must observe that this list of braids is complete (i.e., there are no braids with three strands and up to ten crossings that are not one of these 57). Consider a 3-strand braid T, whose closure is the knot K and the minimum braid representative of K is T^{min}. Since the minimum braid representative minimizes first the number of crossings and, just after that, minimizes the number of strands, it could happen that the number of strands of T^{min} was greater than three and, in this case, there would be no braid among the 57 listed whose closure was K. However, this does not happen because it is known that, for all the knots with up to ten crossings, the number of strands of the minimum braid representative is equal to the smallest number of strands in a braid whose closure is that knot (see [1]) - except for the knot 10_{136}, which is not relevant in our work, since the minimum braid representative has five strands and it can be represented by a braid with, at least four strands.

Below are shown the results of the comparison of the first and second traces of the minimum braid representation of the knots with up ten crossings, whose representation is by a braid with three strands.

Table 3.2: Knots with up to seven crossings

knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	$\mathcal{H}_{\mathcal{G}(2,3)}$	$\mathcal{H}_{\mathcal{G}(2,4)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	$\mathcal{H}_{\mathcal{G}(2,3)}$	$\mathcal{H}_{\mathcal{G}(2,4)}$
3_1	96	4320	24576	$c(3_1')$	96	4752	27648
4_1	48	3024	15360	$c(4_1')$	48	3024	15360
5_2	24	864	1536	$c(5_2')$	24	864	1536
6_2	24	1296	1536	$c(6_2')$	24	1296	1536
6_3	24	864	1536	$c(6_3')$	24	864	1536
7_3	48	1728	–	$c(7_3')$	48	2160	–
7_5	24	432	1536	$c(7_5')$	24	432	1536

Table 3.3: Knots with eight crossings

knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	$\mathcal{H}_{\mathcal{G}(2,3)}$	$\mathcal{H}_{\mathcal{G}(2,4)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	$\mathcal{H}_{\mathcal{G}(2,3)}$	$\mathcal{H}_{\mathcal{G}(2,4)}$
8_2	24	864	1536	$c(8_2')$	24	864	1536
8_5	144	5616	–	$c(8_5')$	144	7776	–
8_7	24	432	1536	$c(8_7')$	24	432	1536
8_9	24	432	1536	$c(8_9')$	24	432	1536
8_{10}	144	7344	–	$c(8_{10}')$	144	9504	–
8_{16}	24	432	–	$c(8_{16}')$	24	432	–
8_{17}	24	864	–	$c(8_{17}')$	24	864	–
8_{18}	336	66096	–	$c(8_{18}')$	336	69120	–
8_{19}	144	9072	–	$c(8_{19}')$	144	9936	–
8_{20}	144	5616	–	$c(8_{20}')$	144	7776	–
8_{21}	144	6048	–	$c(8_{21}')$	144	8208	–

Table 3.4: Knots with nine crossings

knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	$\mathcal{H}_{\mathcal{G}(2,3)}$	$\mathcal{H}_{\mathcal{G}(2,4)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	$\mathcal{H}_{\mathcal{G}(2,3)}$	$\mathcal{H}_{\mathcal{G}(2,4)}$
9_3	24	864	–	$c(9_3')$	24	864	–
9_6	96	4752	–	$c(9_6')$	96	5184	–
9_9	24	432	–	$c(9_9')$	24	432	–
9_{16}	144	7776	–	$c(9_{16}')$	144	8640	–

Table 3.5: Knots with ten crossings

knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	$\mathcal{H}_{\mathcal{G}(2,3)}$	$\mathcal{H}_{\mathcal{G}(2,4)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	$\mathcal{H}_{\mathcal{G}(2,3)}$	$\mathcal{H}_{\mathcal{G}(2,4)}$
10_2	48	864	–	$c(10'_2)$	48	1296	–
10_5	96	4752	–	$c(10'_5)$	96	5184	–
10_9	96	6048	–	$c(10'_9)$	96	6480	–
10_{17}	48	2592	–	$c(10'_{17})$	48	3024	–
10_{46}	24	432	–	$c(10'_{46})$	24	432	–
10_{47}	24	864	–	$c(10'_{47})$	24	864	–
10_{48}	24	1296	1536	$c(10'_{48})$	24	1296	1536
10_{62}	144	9936	–	$c(10'_{62})$	144	13824	–
10_{64}	144	11232	–	$c(10'_{64})$	144	13824	–
10_{79}	24	864	–	$c(10'_{64})$	24	864	–
10_{82}	144	7344	–	$c(10'_{82})$	144	9504	–
10_{85}	144	6912	–	$c(10'_{85})$	144	9072	–
10_{91}	24	432	–	$c(10'_{91})$	24	432	–
10_{94}	24	432	–	$c(10'_{94})$	24	432	–
10_{99}	528	128304	–	$c(10'_{99})$	528	130896	–
10_{100}	24	864	–	$c(10'_{100})$	24	864	–
10_{104}	48	1296	–	$c(10'_{104})$	48	1728	–
10_{106}	96	8640	–	$c(10'_{106})$	96	9072	–
10_{109}	24	864	–	$c(10'_{109})$	24	864	–
10_{112}	96	4320	–	$c(10'_{112})$	96	4752	–
10_{116}	24	432	–	$c(10'_{116})$	24	432	–
10_{118}	24	432	–	$c(10'_{118})$	24	432	–
10_{123}	24	9072	–	$c(10'_{123})$	24	9072	–
10_{124}	24	432	–	$c(10'_{124})$	24	432	–
10_{125}	24	864	–	$c(10'_{125})$	24	864	–
10_{126}	24	432	–	$c(10'_{126})$	24	432	–
10_{127}	24	432	–	$c(10'_{127})$	24	432	–
10_{139}	144	9936	–	$c(10'_{139})$	144	13824	–
10_{141}	144	12960	–	$c(10'_{141})$	144	15552	–
10_{143}	144	11664	–	$c(10'_{143})$	144	15552	–
10_{148}	24	1728	–	$c(10'_{148})$	24	1728	–
10_{149}	24	432	–	$c(10'_{149})$	24	432	–
10_{152}	24	432	–	$c(10'_{152})$	24	432	–
10_{155}	24	432	–	$c(10'_{155})$	24	432	–
10_{157}	48	2590	–	$c(10'_{157})$	48	3024	–
10_{159}	96	7344	–	$c(10'_{159})$	96	7776	–
10_{161}	24	1728	–	$c(10'_{161})$	24	1728	–

Observing these tables, we conclude that the invariant, with $\mathcal{G} = \mathcal{G}(2,2)$ does not distinguishes any of the pairs. However, when $\mathcal{G} = \mathcal{G}(2,3)$ it tells apart about 48% of the pairs, which means that in 48% of the cases it sees beyond the knot group.

Next we present the tables with the same comparison but for the minimum braid representation of links with up to ten crossings. For these, the invariant considering the crossed module $\mathcal{G} = \mathcal{G}(2,2)$ is powerful enough to distinguish all the pairs.

Table 3.6: Links with up to seven crossings

knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$
$L5a1$	336	$c(L5a1')$	756
$L6a4$	3168	$c(L6a4')$	9504
$L6n1$	1032	$c(L6n1')$	2700
$L7a1$	768	$c(L7a1')$	1728
$L7a3$	384	$c(L7a3')$	864
$L7a6$	144	$c(L7a6')$	432
$L7n1$	312	$c(L7n1')$	756
$L7n2$	336	$c(L7n2')$	756

Table 3.7: Links with eight crossings

knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$
$L8a16$	2688	$c(L8a16')$	8424	$L8a18$	1032	$c(L8a18')$	2700
$L8a19$	3000	$c(L8a19')$	7128	$L8n3$	1032	$c(L8n3')$	2700
$L8n4$	1272	$c(L8n4')$	3240	$L8n5$	3168	$c(L8n5')$	9504

Table 3.8: Links with nine crossings

knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$
$L9a2$	346	$c(L9a2')$	756	$L9a9$	1056	$c(L9a9')$	2376
$L9a14$	528	$c(L9a14')$	1188	$L9a20$	240	$c(L9a20')$	648
$L9a21$	120	$c(L9a21')$	378	$L9a22$	120	$c(L9a22')$	378
$L9a28$	288	$c(L9a28')$	1134	$L9a29$	264	$c(L9a29')$	702
$L9a31$	384	$c(L9a31')$	972	$L9a36$	288	$c(L9a36')$	702
$L9a38$	456	$c(L9a38')$	1026	$L9a39$	384	$c(L9a39')$	918
$L9a41$	1104	$c(L9a41')$	2538	$L9n4$	312	$c(L9n4')$	756
$L9n5$	384	$c(L9n5')$	864	$L9n6$	768	$c(L9n6')$	1728
$L9n13$	624	$c(L9n13')$	1512	$L9n14$	264	$c(L9n14')$	702
$L9n15$	288	$c(L9n15')$	702	$L9n16$	288	$c(L9n16')$	702
$L9n17$	264	$c(L9n17')$	702	$L9n18$	1080	$c(L9n18')$	2484
$L9n19$	432	$c(L9n19')$	1026				

Table 3.9: Links with ten crossings

knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$	knot	$\mathcal{H}_{\mathcal{G}(2,2)}$
$L10a138$	3360	$c(L10a138')$	10152	$L10a140$	5904	$c(L10a140')$	14364
$L10a145$	1032	$c(L10a145')$	2700	$L10a148$	1464	$c(L10a148')$	3672
$L10a156$	2112	$c(L10a156')$	5130	$L10a161$	1584	$c(L10a161')$	4914
$L10a162$	2424	$c(L10a162')$	8262	$L10a163$	2952	$c(L10a163')$	9234
$L10n77$	1032	$c(L10n77')$	2700	$L10n78$	1032	$c(L10n78')$	2700
$L10n79$	2688	$c(L10n79')$	8424	$L10n81$	3000	$c(L10n81')$	7128
$L10n92$	1584	$c(L10n92')$	4914	$L10n93$	1824	$c(L10n93')$	5400
$L10n94$	3336	$c(L10n94')$	8316				

Conclusions

In this work, we proposed to compute the crossed module invariant of welded virtual knots arising from welded braids with three strands and up to ten (non virtual) crossings. The calculations were done for three crossed modules - $\mathcal{G}(2,2)$, $\mathcal{G}(2,3)$ and $\mathcal{G}(2,4)$. The invariant with $\mathcal{G} = \mathcal{G}(2,4)$ was only computed for a few examples due to the limitations of time available for this work.

Analyzing the results for the braids for which the closure (using the first trace) is a knot we can see that the crossed module invariant with $\mathcal{G} = \mathcal{G}(2,2)$ does not distinguish any of the pairs obtained by closing the braids in the two different ways. However, when $\mathcal{G} = \mathcal{G}(2,3)$ it is already possible to see the non triviality of the invariant. In this case we were able to distinguish around 48% of the pairs. Unfortunately, the few pairs for which the crossed module invariant was calculated with $\mathcal{G} = \mathcal{G}(2,4)$ remained undistinguished by this invariant.

The example presented beyond the set of examples already mentioned, the welded virtual knot with trivial knot group, was not distinguished by the crossed module invariant when $\mathcal{G} = \mathcal{G}(2,2)$, $\mathcal{G}(2,3)$ and $\mathcal{G}(2,4)$. As this thesis was being concluded, we were still waiting for the results for $\mathcal{G} = \mathcal{G}(2,5)$.

With all these examples we conclude that the crossed module invariant is relatively strong, seeing beyond the knot group in a large number of cases.

Future work could be done by changing the implementation of the program. In particular, it could be changed with the aim of computing it in parallel making the calculations run faster. Another goal is to generalize the implementation so that it accepts braids with more strands and other crossed modules.

Bibliography

[1] Knot atlas. http://katlas.org/.

[2] C. C. Adams. *The Knot Book*. American Mathematical Society, 1994.

[3] R. Fenn. A strange torus embedding.

[4] R. Fenn, R. Rimányi, and C. Rourke. The braid-permutation group. *Topology*, 36, no.1:123–135, 1997.

[5] T. A. Gittings. Minimum braids: A complete invariant of knots and links. arXiv:math/0401051v1, 2004.

[6] S. Kamada. Braid presentation of virtual knots and welded knots. arxiv.org/abs/math/0008092, 2000.

[7] L. H. Kauffman. *Knots and Physics*. World Scientific, 1991.

[8] L. H. Kauffman. Virtual knot theory. *European Journal of Combinatorics*, 20:663–691, 1999.

[9] L. H. Kauffman and S. Lambropoulou. Virtual braids and the l-move. *Journal of Knot Theory and Its Ramifications*, Vol. 15, no. 6:773–811, 2006.

[10] L. H. Kauffman and J. F. Martins. Invariants of welded virtual knots via crossed module invariants of knotted surfaces. arxiv.org/abs/0704.1246, 2008.

[11] C. Livingston. *Knot Theory*. The Mathematical Association of America, 1993.

[12] J. F. Martins. Categorical groups, knots and knotted surfaces. *Journal of Knot Theory and Its Ramifications*, Vol. 16, no. 9:1181–1217, 2007.

[13] S. Satoh. Virtual knot presentation of ribbon torus-knots. *Journal of Knot Theory and Its Ramifications*, Vol. 9, no. 4:531–542, 2000.

[14] A. Sossinsky. *Knots: Mathematics with a Twist*. Harvard University Press, 2002.

Algorithm

```
GL[n_, p_]:=Module[{mat, res = {}, i}, mat = Tuples[Tuples[Range[0, p − 1], n], n];
        i = 1;
        If[p ≠ 4,
            While[i ≤ Length[mat],
                    If[Mod[Det[mat[[i]]], p] == 0, i++, res = Append[res, mat[[i]]]; i++]],
            While[i ≤ Length[mat],
                    If[Mod[Det[mat[[i]]], p] == 0||Mod[Det[mat[[i]]], p] == 2, i++,
                        res = Append[res, mat[[i]]]; i++]]; ];
        res]

Vect[n_, p_]:=Module[{v}, v = Tuples[Range[0, p − 1], n];
        v]

f[braid_, n_, p_, col_]:=Module[{i, res = {}, l = {}, c, a},
                    res = col;
                    For[i = 1, i ≤ Length[braid[[2]]], i++,
                        c = braid[[2, i]];
                        l = res;
                        Which[!IntegerQ[c], a = ToExpression[StringDrop[ToString[c], {1}]];
                                        res[[a]] = l[[a + 1]];
                                        res[[a + 1]] = l[[a]],
                            c < 0, res[[−c]] = {l[[−c + 1, 1]],
Apply[Function[{y, e, f}, Mod[−Inverse[y, Modulus → p].e + e + f, p]], {l[[−c + 1, 1]], l[[−c, 2]], l[[−c + 1, 2]]}]};
                                    res[[−c + 1]] =
{Apply[Function[{x, y}, Mod[Inverse[y, Modulus → p].x.y, p]], {l[[−c, 1]], l[[−c + 1, 1]]}],
Apply[Function[{y, e}, Mod[Inverse[y, Modulus → p].e, p]], {l[[−c + 1, 1]], l[[−c, 2]]}]},
                            c > 0, res[[c]] =
{Apply[Function[{x, y}, Mod[x.y.Inverse[x, Modulus → p], p]], {l[[c, 1]], l[[c + 1, 1]]}],
Apply[Function[{x, f}, Mod[x.f, p]], {l[[c, 1]], l[[c + 1, 2]]}]};
                                    res[[c + 1]] = {l[[c, 1]],
Apply[Function[{x, e, f}, Mod[Mod[−x.f, p] + e + f, p]], {l[[c, 1]], l[[c, 2]], l[[c + 1, 2]]}]}]; ];
                    res]

conta[braid_, n_, p_]:=Module[{i, j, k, pr = {}, c = {}, r, res = 0},
                    pr = Tuples[{GL[n, p], Vect[n, p]}];
                    For[i = 1, i ≤ Length[pr], i++,
                        For[j = 1, j ≤ Length[pr], j++,
                            For[k = 1, k ≤ Length[pr], k++,
                            c = {pr[[i]], pr[[j]], pr[[k]]};
                            r = f[braid, n, p, c];
                            If[r===c, res++]; ]; ]; ];
                    res]
```

```
conta2[braid_, n_, p_]:=Module[{pr, pf, i, j, k, c, r, res = 0},
                    pr = Tuples[{GL[n, p], Vect[n, p]}];
                    pf = Tuples[{GL[n, p], {Vect[n, p][[1]]}}];
                    For[i = 1, i ≤ Length[pr], i++,
                        For[j = 1, j ≤ Length[pr], j++,
                            For[k = 1, k ≤ Length[pf], k++,
                                c = {pr[[i]], pr[[j]], pf[[k]]};
                                r = f[braid, n, p, c];
                                If[r===c, res++]; ]; ]; ];
                    res = res * p^n;
                    res]
```

Example For the minimum braid representation of the knot 6_2, we have:

```
conta[{3, {−1, −1, −1, 2, −1, 2}}, 2, 2]

24

conta2[{3, {−1, −1, −1, 2, −1, 2}}, 2, 2]

24

conta[{3, {−1, −1, −1, 2, −1, 2}}, 2, 3]

1296

conta2[{3, {−1, −1, −1, 2, −1, 2}}, 2, 3]

1296
```

And for the minimum braid representation of the link $L5a1$:

```
conta[{3, {−1, 2, −1, 2, −1}}, 2, 2]

336

conta[{3, {−1, 2, −1, 2, −1}}, 2, 2]

756
```