Fall 2010

Calculating knot distances and solving tangle equations involving Montesinos links

Hyeyoung Moon
University of Iowa

CALCULATING KNOT DISTANCES AND SOLVING TANGLE EQUATIONS INVOLVING MONTESINOS LINKS

by

Hyeyoung Moon

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Applied Mathematical and Computational Sciences
in the Graduate College of
The University of Iowa

December 2010

Thesis Supervisor: Associate Professor Isabel K. Darcy

ABSTRACT

My thesis is divided into two parts : (1) improving the knot distance table theoretically and computationally and (2) solving tangle equations involving Montesinos links.

The knot distance between two knots is defined as the minimum number of crossing changes required to convert one knot to the other. Topoisomerases are enzymes involved in changing crossings of DNA knots. Thus the study of knot distances can be used to study topoisomerase action. Using some mathematical theories, knot distances have been tabulated for rational knots, composites of rational knots up to 13 crossings and 8 crossing nonrational prime knots. However, there are still undetermined distances in the knot distance table. In this thesis, the Jones polynomial and the signature of knots are used to improve lower bounds of knot distances.

Proteins bind to DNA segments to catalyze several biological processes. Such protein-DNA complexes can be modeled using tangles. A tangle is a 3-dimensional ball with strings properly embedded in it. A tangle model assumes that the protein is a 3-dimensional ball and the DNA segments bound by the protein are strings embedded inside the ball. Enzyme actions can change the topology of DNA within an enzyme-DNA complex. Thus enzyme actions can be modeled as tangle equations. The goal is to determine possible topological configurations of DNA within the enzyme-DNA complex by solving a system of tangle equations. In this thesis, we solve systems of tangle equations when substrate DNA and product DNA are assumed to be either Montesinos links or rational knots. We assume that at least one of them is a Montesinos link.

Abstract Approved: ______________________________
Thesis Supervisor

____
Title and Department

____
Date

CALCULATING KNOT DISTANCES AND SOLVING TANGLE EQUATIONS INVOLVING MONTESINOS LINKS

by

Hyeyoung Moon

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Applied Mathematical and Computational Sciences
in the Graduate College of
The University of Iowa

December 2010

Thesis Supervisor: Associate Professor Isabel K. Darcy

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Hyeyoung Moon

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Applied Mathematical and Computational Sciences at the December 2010 graduation.

Thesis Committee: ______________________________
Isabel K. Darcy, Thesis Supervisor

____
Dennis Roseman

____
Jonathan Simon

____
Colleen Mitchell

____
Rodica Curtu

ACKNOWLEDGEMENTS

I cannot say enough thanks to all people who have helped and supported me in various ways to finish my dissertation. First and foremost, I would like to thank God for guiding me and giving me strength to go through this long journey.

I am heartily grateful to my advisor, Dr. Isabel K. Darcy for her profuse guidance, encouragement and warm concern throughout the long years of my study. I have been motivated a lot by her positive attitude and perpetual energy toward the research.

A generous support from Mathematics and AMCS department at the University of Iowa is greatly appreciated. Special thanks to Dr. Herbert Hethcote, Dr. Yi Li and Dr. Weimin Han for their thoughtful considerations as AMCS chairs.

I want to express my thanks to my math teachers throughout my life who gave me enthusiasm about mathematics.

I am indebted to my friends and colleagues who have supported me mathematically, morally and with many other things. I also would like to thank those who pray for me.

My deepest gratitude goes to my parents and family in Korea for their unflagging love, support and prayer throughout my life. Lastly, and most importantly, I would like to thank my husband Yangho, my kids Elaine and Daniel for their unconditional love, support and understanding of my absence.

This research was supported in part by a grant from the Joint DMS/NIGMS

Initiative to Support Research in the Area of Mathematical Biology (NSF 0800285).

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

ABSTRACT

My thesis is divided into two parts : (1) improving the knot distance table theoretically and computationally and (2) solving tangle equations involving Montesinos links.

The knot distance between two knots is defined as the minimum number of crossing changes required to convert one knot to the other. Topoisomerases are enzymes involved in changing crossings of DNA knots. Thus the study of knot distances can be used to study topoisomerase action. Using some mathematical theories, knot distances have been tabulated for rational knots, composites of rational knots up to 13 crossings and 8 crossing nonrational prime knots. However, there are still undetermined distances in the knot distance table. In this thesis, the Jones polynomial and the signature of knots are used to improve lower bounds of knot distances.

Proteins bind to DNA segments to catalyze several biological processes. Such protein-DNA complexes can be modeled using tangles. A tangle is a 3-dimensional ball with strings properly embedded in it. A tangle model assumes that the protein is a 3-dimensional ball and the DNA segments bound by the protein are strings embedded inside the ball. Enzyme actions can change the topology of DNA within an enzyme-DNA complex. Thus enzyme actions can be modeled as tangle equations. The goal is to determine possible topological configurations of DNA within the enzyme-DNA complex by solving a system of tangle equations. In this thesis, we solve systems of tangle equations when substrate DNA and product DNA are assumed to

be either Montesinos links or rational knots. We assume that at least one of them is a Montesinos link.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
INTRODUCTION

1.1 Biological background

Biological processes are influenced by DNA topology and can change topological properties of DNA. Proteins act on DNA molecules to solve topological problems which occur during biological processes such as replication, transcription and recombination. There are two main types of enzymes involved in these processes: recombinases and topoisomerases. *Recombinases* bind and cut two DNA segments, and interchange and reseal the ends in different ways. Thus these enzymes are involved in the genetic exchange of DNA. *Topoisomerases* are enzymes which break one or two strands of DNA, pass another segment of DNA through the break, and rejoin the break. These enzymes can change the topology of circular DNA by strand passages.

1.2 Knot distances

Stand passages by topoisomerases correspond to crossing changes of knots/links. The distance between two knots is defined as the minimum number of crossing changes needed to covert one knot to the other. Thus knot distances can be used to study topoisomerase action. Isabel K. Darcy calculated knot distances for rational knots and composites of rational knots up to 13 crossings using computer programming based on mathematical theories. Eight crossing nonrational prime knots were input by hand [10, 18]. However, there are still undetermined values in the knot distance table. In this thesis, lower bounds of knot distances are improved theoretically and computationally using the Jones polynomial and the signature. A. Stoimenow introduced relations between signed unknotting number one knots and the Jones polynomial in [48]. The unknotting number is a special case of the knot distance. So we generalize it to the case for signed knot distances which can be used to improve knot distances whose lower bounds are one. Properties of the Jones polynomial are employed to check whether lower bounds of knot distances can be changed from 1 to 2. For computational purposes, computer programming involving C/C++ and MATLAB are utilized to generate a new knot distance table with improved lower bounds.

1.3 Solving tangle equations

During biological processes, proteins (enzymes) bind to DNA to fulfill their roles. Protein interaction with DNA can change the topology of DNA which results in knotted or linked DNA. Biologists are interested in identifying the topological configuration of DNA to study enzyme mechanisms. The topology of DNA can sometimes be determined using biological methods such as cryo-electron microscopy, AFM (Atomic Force Microscopy) or crystal structures of small proteins. However, it is a difficult and laborious process which often doesn't work. Thus tangle analysis was introduced to study various enzyme actions mathematically. C. Ernst and D. W. Sumners first used tangles to model protein-DNA complexes [20]. A tangle is a 3-dimensional ball with strings properly embedded in it. In the tangle model, we assume the protein complex as a 3-dimensional ball and the DNA segments bound by protein as strings embedded inside the ball. An enzyme action can be modeled by replacing one tangle with another. This gives a system of tangle equations. Solving this system of tangle equations can help determine the pathways of enzyme actions and possible topological configurations of DNA bound by protein. We solve the system of tangle equations assuming that the topology of DNA before and after enzyme action is either Montesinos links or rational links and at least one of them is a Montesinos link. Relations between rational tangle replacement and surgery on the double branch cover of a substrate knot are described.

1.4 Organization of thesis

In chapter 2, the necessary mathematical definitions and preliminaries are provided. Background in biology and mathematics is given in chapter 3. In chapter 4, we develop theories about the signed knot distances and the Jones polynomial. The methodology and algorithm of how the theories are applied to get new data are explained in section 4.5 and section 4.6. A new distance table is given in appendix C. In chapter 5, we solve systems of tangle equations involving Montesinos links. Section 5.4 is devoted to explain how a rational tangle replacement is related to the surgery on the double branch cover. Conclusion and future direction are stated in chapter 6. Appendices A and B give README files and some programming codes used for knot distance calculation.

CHAPTER 2
DEFINITIONS AND PRELIMINARIES

In this chapter, we describe basic mathematical definitions and preliminaries which will be used throughout this thesis. We are working in the piecewise linear or smooth category.

2.1 Knot theory

Definition 2.1. Let X and Y be topological spaces. A function $h : X \to Y$ is called a *homeomorphism* if h is 1-1, onto, continuous, and h^{-1} is continuous.

Definition 2.2. A function $i : X \to Y$ is an *embedding* if $i : X \to i(X)$ is a homeomorphism.

Definition 2.3. [43] A subset K of a space X is a *knot* if K is homeomorphic with a sphere S^p. More generally K is a *link* if K is homeomorphic with a disjoint union $S^{p_1} \cup \cdots \cup S^{p_r}$ of one or more spheres.

An alternative definition is that a knot is an embedding $K : S^p \to X$. We take X to be $\mathbf{M}^{p+2}$, a manifold of dimension $p + 2$.

Definition 2.4. Let X and Y be two topological spaces. If f_1 and f_2 are continuous maps from X to Y, we say that f_1 is *homotopic* to f_2 if there exists a continuous map $F : X \times [0, 1] \to Y$ such that $F(x, 0) = f_1(x)$ and $F(x, 1) = f_2(x)$ for each $x \in X$. F is called a *homotopy* between f_1 and f_2.

Definition 2.5. A homotopy $F : X \times [0,1] \to X$ is called an *ambient isotopy* if $F(x,0)$ is the identity and $F(x,t)$ is a homeomorphism from X to X for $t \in [0,1]$.

Definition 2.6. A *knot projection* is a projection of a knot $K \subset S^3$ into a 2-dimensional plane where under and over strands are not specified. In this projection, no three points in K correspond to one point on the plane and strands cross transversely.

Definition 2.7. A *knot diagram* is a projection where under and over strands are specified at each crossing as in Figure 2.1 (a).

A knot diagram is *minimal* if it has the minimum number of crossings needed to draw the knot. If a diagram is oriented, we can assign +1 or -1 at each crossing as in Figure2.1 (b).

(a) (b) (c)

Figure 2.1: Partial knot diagram;(a) A crossing of a knot diagram. (b) +1 crossing. (c) -1 crossing.

Definition 2.8. [43] Two knots/links K, $K^{'}$ in X are *ambient isotopic* if there is an ambient isotopy $F_t : X \times [0,1] \to X$ such that $F_1(K) = K^{'}$.

Two knots/links are equivalent (ambient isotopic) in S^3 if and only if two knot diagrams are equivalent in a 2-dimensional plane. Two knot/link diagrams are equivalent if they are related by Reidemeister moves (Figure 2.2).

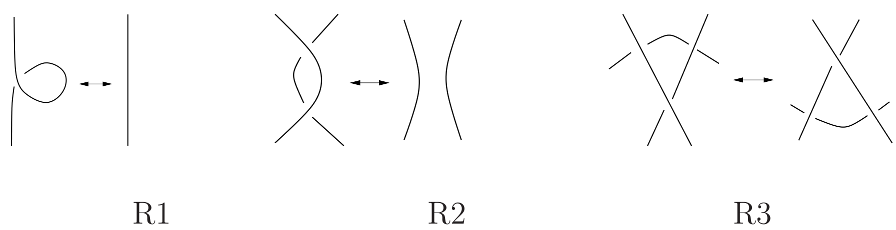


Figure 2.2: Reidemeister moves.

We can compose two knots by removing an arc from each knot and then connecting the four end points by two new arcs as in Figure 2.3. A knot is called a *composite knot* if it can be decomposed into two nontrivial knots. A *prime knot* is a knot which cannot be represented as a composition of two nontrivial knots.

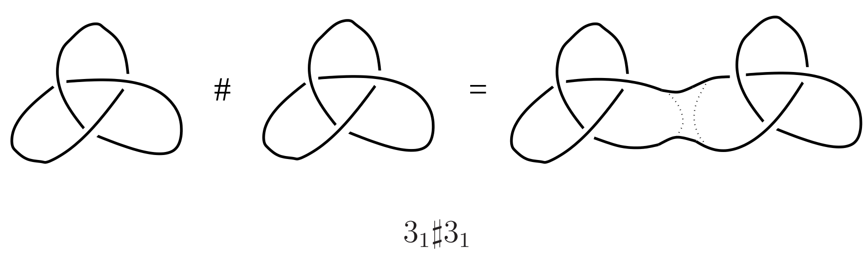
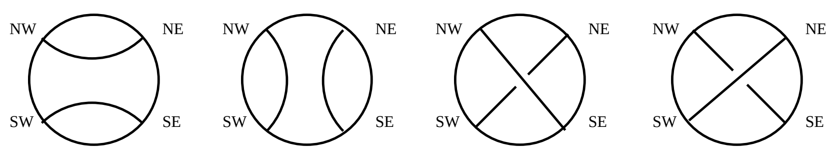


Figure 2.3: Composition of two knots.

2.2 Tangles

Definition 2.9. A *2-string tangle* is a pair (B^3, t), where B^3 is a 3 dimensional ball and t is a pair of arcs properly embedded in B^3. Here, the four endpoints of the arcs are fixed at $NW = (e^{\frac{5\pi i}{4}}, 0)$, $NE = (e^{\frac{\pi i}{4}}, 0)$, $SW = (e^{-\frac{5\pi i}{4}}, 0)$ and $SE = (e^{-\frac{\pi i}{4}}, 0)$.

Examples of 2-string tangles are given in Figure 2.4. A tangle has parity 0 if the arc that starts at NW ends at NE. A tangle has parity ∞ if the arc that starts at NW ends at SW. A tangle has parity 1 if the arc that starts at NW ends at SE.



(0)	$(0, 0)$	(1)	(-1)
0 tangle	∞ tangle	$+1$ tangle	-1 tangle

Figure 2.4: 2-string tangle examples.

Definition 2.10. Two tangles are *equivalent* if they are ambient isotopic keeping the boundary of B^3 fixed.

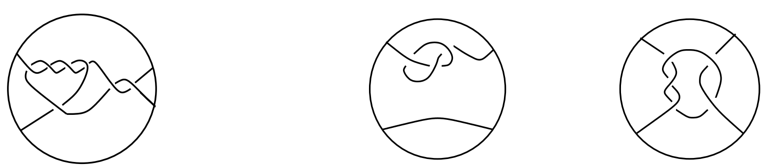
Two tangles are equivalent if and only if two tangle diagrams are equivalent which means that they are related by Reidemeister moves (Figure 2.2).

Definition 2.11. A tangle is *rational* if it is ambient isotopic to the 0 tangle where the boundary of B^3 need not be fixed. Figure 2.5 (a) shows an example of a rational tangle.

There are two types of tangles which are not rational.

Definition 2.12. A tangle is *locally knotted* if there is a 3-ball in B^3 which meets one of two arcs transversely at two points and the arc inside the 3-ball is locally knotted. An arc inside a 3-ball is *locally knotted* if there exists a 2-sphere which meets the arc at two points and there exists another arc on the 2-sphere which connects two end points of the knotted arc and makes a knot combined with the knotted arc. Note that the choice of an arc on a 2-sphere doesn't matter. any arc connecting See example in Figure 2.5 (b).

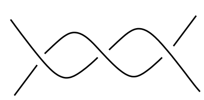
Definition 2.13. A tangle is *prime* if it is neither rational nor locally knotted. See example in Figure 2.5 (c).

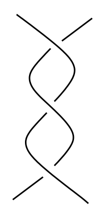


(a) Rational (b) Locally knotted (c) Prime

Figure 2.5: Tangle types.

A rational tangle can be obtained from the 0 tangle or the ∞ tangle by alternating between horizontal half twists and vertical half twists. Horizontal twists represent twists of NE and SE endpoints and vertical twists represent twists of SW and SE endpoints. The rational tangle obtained in this way can be expressed as a vector $(x_1, \cdots, x_n)$ where the numbers alternate between horizontal twists and vertical twists with the last number always representing horizontal twists. So if n is even, we start with vertical twists on the ∞ tangle and if n is odd, we start with horizontal twists on the 0 tangle. Horizontal twists are right-hand twists (left-hand twists) if the corresponding number of twists is positive (negative). Vertical twists are left-hand twists (right-hand twists) if the corresponding number of twists is positive (negative). See Figure 2.6.





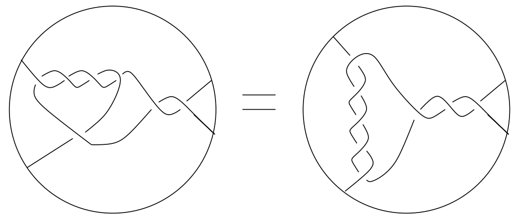
+3 horizontal twists   +3 vertical twists

Figure 2.6: Sign convention for horizontal/vertical twists.

A tangle $(x_1, \cdots, x_n)$ is uniquely identified by its continued fraction, $x_n + \frac{1}{x_{n-1} + \cdots \frac{1}{x_1}}$ [9]. A tangle whose corresponding continued fraction is an integer is called an *integral tangle*. Two rational tangles are equivalent if and only if their continued fractions are the same [9]. For example, the two tangles in Figure 2.7 are equivalent.

Since there are many vectors that have the same continued fractions, the vector

$$2 + \frac{1}{1+\frac{1}{4}} = \frac{14}{5} = 3 + \frac{1}{-4+\frac{1}{-1}}$$

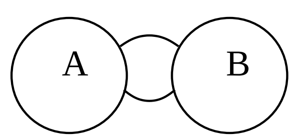


$(4, 1, 2)$ $(-1, -4, 3)$

Figure 2.7: Equivalent tangles.

representation for a tangle is not unique. However, every rational tangle, excluding the four tangles $(0), (\pm 1), (0, 0)$, has a unique canonical form of vector representation $(x_1, \cdots, x_n)$, where $|x_1| > 1, x_i \neq 0$ for $1 \leq i \leq n-1$, and all nonzero x_i's have the same sign [9].

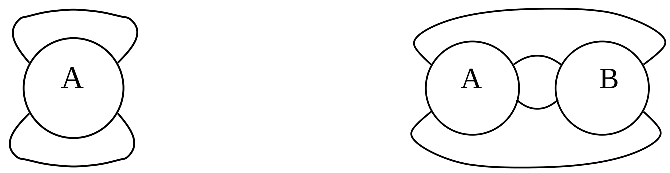
The sum of two tangles A and B, $A + B$ is obtained by connecting NE and SE endpoints of A to NW and SW endpoints of B, respectively (Figure 2.8).



$A + B$

Figure 2.8: Tangle sum.

The numerator closure of a tangle A, $N(A)$ is formed by connecting NW and NE endpoints and SW and SE endpoints (Figure 2.9). The numerator closure of a tangle or the sum of tangles forms a knot or link.



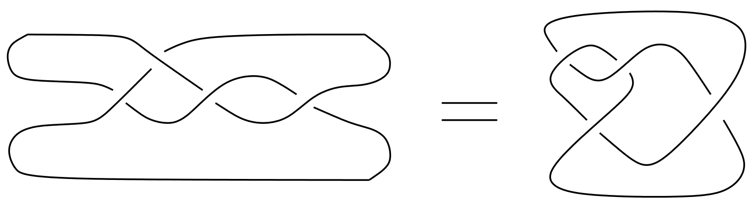
$N(A)$ $\qquad$ $N(A+B)$

Figure 2.9: Numerator closure.

Definition 2.14. [34] A *rational knot*, $N(\frac{a}{b})$, is a knot or link which can be written as the numerator closure of a rational tangle whose corresponding continued fraction is $\frac{a}{b}$.

A rational knot is also called a *4-plat* or *2-bridge knot/link*. A 4-plat $S(a,b)$ is written as $< x_1, \cdots, x_n >$ where $\frac{a}{b} = x_1 + \frac{1}{x_2 + \cdots \frac{1}{x_n}}$. Note that $S(a,b) = N(\frac{a}{-b})$. An example of 4-plat notation of a rational knot is given in Figure 2.10.

Theorem 2.1. *[46, 6] Two unoriented rational knots* $N(\frac{a_1}{b_1})$ *and* $N(\frac{a_2}{b_2})$*,* $a_i \geq 0$ *are the same if and only if* $a_1 = a_2$ *and* $b_1 b_2^{\pm 1} \cong 1$*(mod* a_1*).*



$S(5,3) =< 1,1,2 >$ $\qquad$ $N(\frac{5}{-3}) = N((-2,-1,-1))$

Figure 2.10: 4-plat notation of figure eight knot.

CHAPTER 3
BACKGROUND

3.1 Topoisomerase and knot distance

Circular DNA molecules combined with the double helical structure of DNA impose some constraints on biological mechanisms [2]. Unwinding two strands of double helical DNA whose ends are fixed results in stress that is relieved by supercoiling like a telephone cord. At the end of DNA replication of circular DNA, the product DNA molecules are often linked [2]. Topoisomerases solve these topological problems by cutting one or two DNA strands, passing another DNA strands through a transient break and rejoining the break. See Figure 3.1.

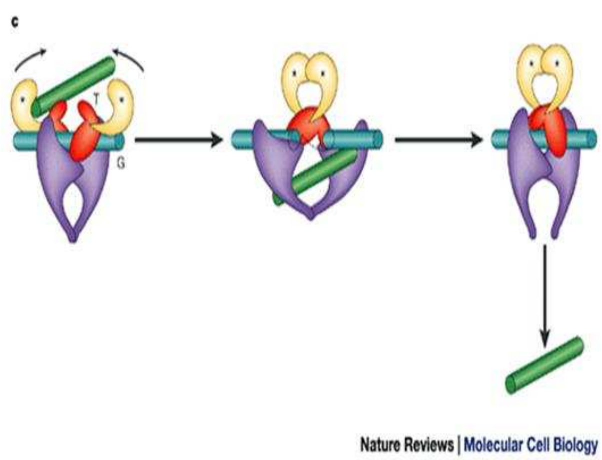


Figure 3.1: A possible mechanism of a type II topoisomerase. This figure is taken from [53] by permission.

There are two types of Topoisomerases, type I and type II [2]: Type I topoisomerase cleaves a single strand of DNA whereas type II topoisomerase cleaves both strands of double stranded DNA. Topoisomerases do not change the chemical structure of DNA but can change the topology of DNA. Type I topoisomerase can change the topology of single stranded DNA and Type II topoisomerase can change the topology of double stranded DNA. Strand passages by topoisomerases can result in knotting/unknotting and linking/unlinking. They are involved in maintaining the proper supercoiling of DNA during DNA replication, transcription and recombination. Topoisomerase are also targets for antibacterial or anticancer drugs [2].

Strand passages by these enzymes correspond to crossing changes of knots (See Figure 3.2). The minimum number of crossing changes needed to convert one knot to another knot is called the *knot distance* between these two knots. Figure 3.3 shows an example when 5_1 is obtained from 5_2 by a single crossing change in a nonminimal diagram of 5_2 [17].

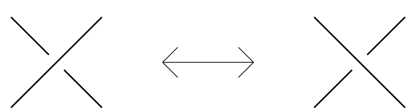


Figure 3.2: Strand passage = crossing change.

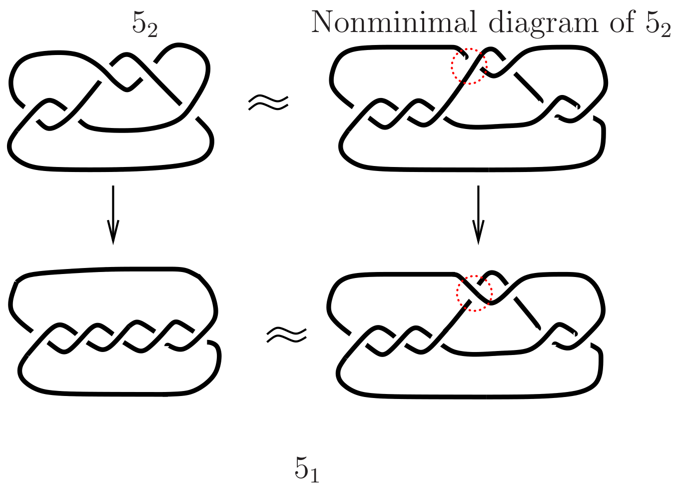


Figure 3.3: Nonminimal diagram example [18].

3.2 Knot distance tabulation

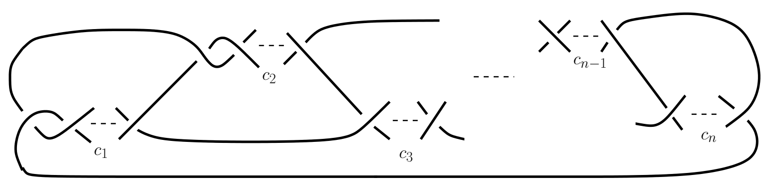
Isabel K. Darcy calculated knot distances for rational knots, composites of rational knots up to 13 crossings and 8 crossing non-rational prime knots [10, 16]. The following mathematics were used to calculate knot distances:

- $d(< c_1, \cdots, c_i, \cdots, c_n >, < c_1, \cdots, c_i - 2, \cdots, c_n >) = 1$,
- Classification of distance one rational knots [16, 49],
- Triangle inequality,
- $d(K_1, K_2) \geq |\dfrac{\sigma(K_1) - \sigma(K_2)}{2}|$ where $\sigma(K)$ is the signature of a knot K [37],
- Unknotting number one knots are prime [45, 54],
- If $d(K, K') = 1$, then there exist $a \in H_1(M_K)$ and $a' \in H_1(M_{K'})$ such that $\lambda(a, a) \equiv n/|H_1(M_K)|$ and $\lambda(a', a') \equiv m/|H_1(M_{K'})| (\text{mod } 1)$ where $m = n =$

$\frac{\pm 1}{2}(|H_1(M_K)|-|H_1(M_{K'})|)$ if $\sigma(K)-\sigma(K')=0$ or ± 2 or $-m=n=\frac{\pm 1}{2}(|H_1(M_K)|$ $+|H_1(M_{K'})|)$ if $\sigma(K)-\sigma(K')=\mp 2$ or 0 and $\lambda : H_1(M_k)\times H_1(M_k)\to \mathbb{Q}/\mathbb{Z}$ is the linking form where M_K is the double branched cover of S^3 over K [37, 10],

- If $d(S(a,b), S(uw_1,v)\sharp S(uw_2,m))=1$, then $u|a$ [10],

- If $d(S(a,b), \sharp_{i=1}^{n} S(uw_i,v))\neq 1$ for $n>2$, $u\neq 1$ [10].



$< c_1, \cdots, c_n >$

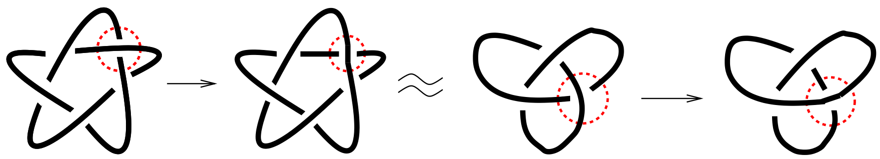
Figure 3.4: 4-plat notation of a rational knot.

The distances between two knots up to mirror images are tabulated. See Table 3.1. However, there are still undetermined values in the knot distance table. An example is given in Figure 3.5.

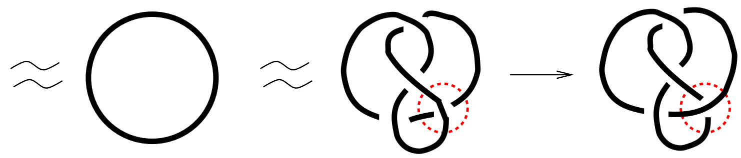
The unknown values can be obtained using (1) mathematical theories, (2) tangle tabulation for distance one knots and (3) experimental and computational data which determine upper bounds of knot distances. One can determine upper bounds of knot distances between two knots simply by changing crossings of one knot from a certain knot diagram until one gets to the other knot. Upper bounds of unknown knot distances also can be determined using computational data. A.

Table 3.1: Old knot distance table [10, 18].

	0_1	3_1	4_1	5_1	5_2	6_1	6_2	6_3	$3_1 \# 3_1$	$3_1 \# 3_1^*$	7_1
3_1	1	0	2	1	1	2	1	1	1	1	2
3_1^*	1	2	2	3	2	2	2	1	3	1	4
4_1	1	2	0	2-3	2	1	1	2	2-3	2-3	3-4
5_1	2	1	2-3	0	1	2-3	2	2	2	2	1
5_1^*	2	3	2-3	4	3	2-3	3	2	4	2	5
5_2	1	1	2	1	0	2	2	2	2	2	2
5_2^*	1	2	2	3	2	2	2	2	3	2	4
6_1	1	2	1	2-3	2	0	1	2	2-3	1-3	3-4
6_1^*	1	2	1	2-3	2	1	2	2	2-3	1-3	3-4
6_2	1	1	1	2	2	1	0	2	2	2	2-3
6_2^*	1	2	1	3	2	2	2	2	3	2	4
6_3	1	1	2	2	2	2	2	0	2	2	3
$3_1 \# 3_1$	2	1	2-3	2	2	2-3	2	2	0	2	2-3
$3_1^* \# 3_1^*$	2	3	2-3	4	3	2-3	3	2	4	2	5
$3_1 \# 3_1^*$	2	1	2-3	2	2	1-3	2	2	2	0	3
7_1	3	2	3-4	1	2	3-4	2-3	3	2-3	3	0
7_1^*	3	4	3-4	5	4	3-4	4	3	5	3	6
7_2	1	2	2	2	1	2	2	2	2-3	2-3	2
7_2^*	1	2	2	3	2	2	2	2	3	2-3	4
7_3	2	3	2-3	4	3	2-3	3	2-3	4	2-3	5
7_3^*	2	2	2-3	1	1	2-3	2-3	2-3	2-3	2-3	1
7_4	2	2-3	2-3	3-4	2-3	2-3	2-3	2	3-4	2	4-5
7_4^*	2	1	2-3	2	1	2-3	2	2	2	2	2
7_5	2	1	2-3	1	1	2-3	2	2	2	2	1
7_5^*	2	3	2-3	4	3	2-3	3	2	4	2	5
7_6	1	1	1	2	1	2	2	2	2	2	2-3
7_6^*	1	2	1	3	2	2	2	2	3	2	4
7_7	1	2	1	2-3	2	2	2	2	2-3	1-2	3-4
7_7^*	1	1	1	2	2	2	2	2	2	1-2	3
$3_1 \# 4_1$	2	1	1	1-2	1-2	2	2	2	2	2	2-3
$3_1^* \# 4_1$	2	2-3	1	3-4	2-3	2	2	2	3-4	2	4-5
8_1	1	2	2	2-3	2	1	2	2	2-3	2-3	3-4
8_1^*	1	2	2	2-3	2	2	2	2	2-3	2-3	3-4
8_2	2	1	2	1	2	2	1	2	2	2	2
8_2^*	2	3	2	4	3	2-3	3	2	4	2	5
8_3	2	2-3	2	2-4	2-3	1	2	2-3	2-4	2-4	3-5
8_4	2	2	1	2-3	2-3	2	1	2-3	2-3	2-3	2-4
8_4^*	2	2-3	1	3-4	2-3	1	2	2-3	3-4	2-3	4-5
8_5	2	3	2	4	3	2-3	3	2-3	4	2-3	5
8_5^*	2	1-2	2	2-3	1-3	2	1	2-3	1	2-3	1-4



5_1 3_1



0_1 4_1

$$d(5_1, 4_1) \leq 3.$$

$$d(5_1, 4_1) \geq \frac{1}{2}|\sigma(5_1) - \sigma(4_1)| = \frac{1}{2}|4 - 0| = 2.$$

Figure 3.5: Unknown knot distance.

Flammini, A. Maritan and A. Stasiak determined some distances via computational software modeling biological reactions [23]. Ram K. Medikonduri, Melanie DeVries and Danielle Washburn have tabulated realizable (drawable) tangles. The pair of knots $N(T+1)$ and $N(T-1)$ has distance one when T is a tangle. Hence such distance one pairs can be created using a table of tangles. The work on generating such knot pairs and identifying them by using the HOMFLYPT polynomial is in progress. We expect to detect new pairs of knots whose distances are one.

3.3 Recombination

Recombination refers to the process of the genetic rearrangement of DNA. Enzymes which carry out this process are called *recombinases*. A recombination reaction which occurs only at very specific sites on DNA segments is called *site-specific recombination*. The specific DNA sequences are called the *target sites*.

Site specific recombination can result in inversion, deletion or insertion of DNA sequences depending on how target sites are oriented. If two target sites are oriented oppositely to each other (inverted repeat) in the same DNA molecule, a DNA sequence is inverted with respect to the other (Figure 3.6(a)). If two target sites are oriented in the same direction (direct repeat) in the same DNA molecule, a DNA sequence is deleted from the rest of DNA (Figure 3.6(b)). If a site-specific recombination takes place on target sites from two different DNA molecules, one DNA sequence is inserted into the other.

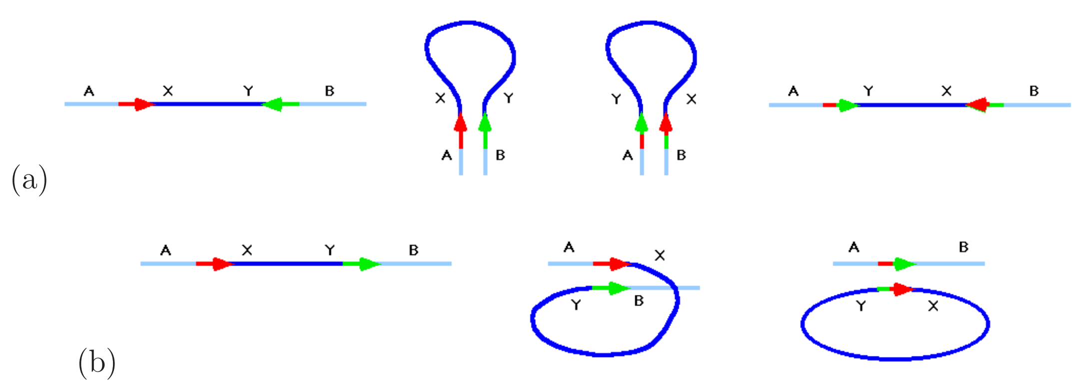
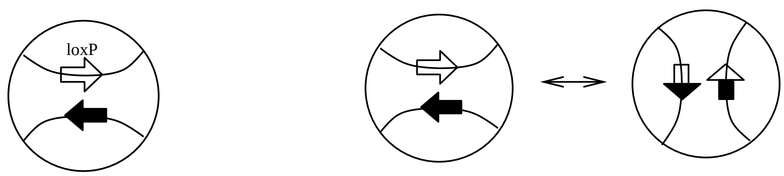


Figure 3.6: Site specific recombination;(a) Inversion. (b) Deletion/Insertion. (This figure is redrawn from http://www.mun.ca/biochem/courses/3107/Lectures/To -pics/Site-specific-Recomb.html.)

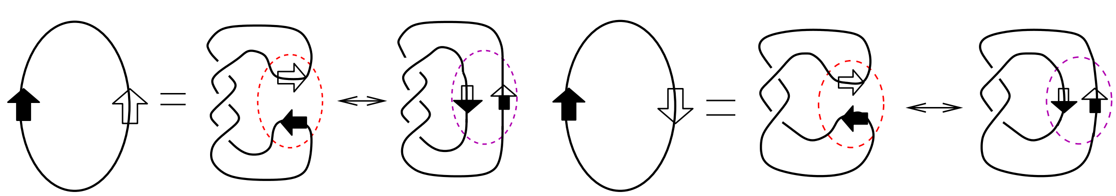
Cre recombinase catalyzes site-specific recombination on target site, called *loxP*. According to the *Cre* model in [28], *Cre* can only act when *loxP* sites are in anti-parallel orientation in the *Cre*-DNA complex (See Figure 3.7(a)). Figure 3.7(b) shows how *Cre* acts on target sites locally. Inversion and deletion by *Cre* on a circular DNA molecule produces a knot and a link, respectively. See Figure 3.8.

Flp recombination is analogous to *Cre* recombination. *Flp* recombinases mediate site-specific recombination on target site, called *FRT* [55].



(a) (b)

Figure 3.7: *Cre* recombination on *loxP*;(a) Antiparallel orientation of *loxP*. (b) Mechanism of *Cre* recombinase.

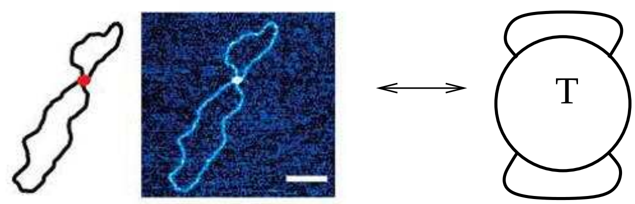


(a) (b)

Figure 3.8: Knots and links produced by *Cre* recombintation; (a) Inversion and (b) Deletion by *Cre* on circular DNA [13].

3.4 Tangle equation

A *2-string tangle* is a 3-dimensional ball with two strings properly embedded in it as in Figure 2.4. Protein-DNA complexes can be modeled using 2-string tangles [20]. In a tangle model, protein which binds two DNA strands is considered as a 3-dimensional ball and the two DNA strands bound by the protein as strings inside the ball. In Figure 3.9 (a), a protein-DNA complex is shown as a blob and two DNA loops not bound by the protein are coming out of it. Figure 3.9 (b) shows a corresponding tangle model.



(a) (b)

Figure 3.9: Tangle model for recombination;(a) AFM image of a Flp (recombinase) complex formed with circular DNA [52] and (b) a corresponding tangle model.

Enzymes bind to segments of the starting DNA (substrate DNA). Enzyme actions can change the topology of DNA within these protein-DNA complexes. Tangle models assume that topological changes occur locally and thus the topology of the DNA outside the protein-DNA complex is not affected by enzyme action. If we also represent the DNA outside the protein-DNA complex as a tangle, we have a tangle equation as the topology of the substrate DNA equals the numerator closure of the sum of the DNA outside the protein-DNA complex and the protein-DNA complex.

Let the tangles B and E represent protein-bound DNA complexes before and after enzyme action, respectively and tangle U be the DNA not bound by protein. Then enzyme action can be modeled by replacing the tangle B with the tangle E. U combined with B and E gives us the topology of the substrate DNA and the product DNA, respectively. For example, Figure 3.10 formulates topoisomerase action on a trefoil knot as a system of tangle equations. The red dotted circle with arcs ((-1) tangle) is an example of B and the purple dotted circle with arcs ((+1) tangle) is an example of E. By replacing the (-1) tangle with the (+1) tangle, we have the unknot. Thus we can conclude that topoisomerase action on a trefoil knot can produce the unknot. Tangle equations modeling topoisomerase action are given in Figure 3.11.

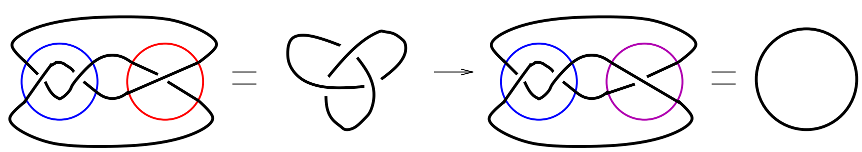


Figure 3.10: A model of a topoisomerase action [13].

Substituting the red dotted circle with the purple dotted circle (or vice versa) in Figure 3.8(a) and (b) models *Cre* recombination. Figure 3.12 shows an example of tangle equations modeling *Cre* recombination [13]. There are various tangle models for recombinases using various mechanisms.

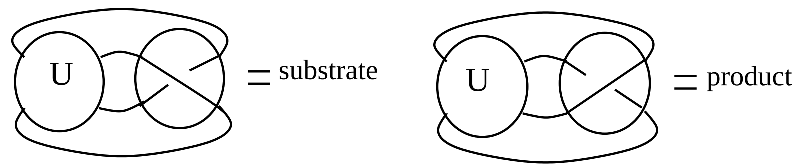


Figure 3.11: Tangle equations modeling topoisomerase action [13].

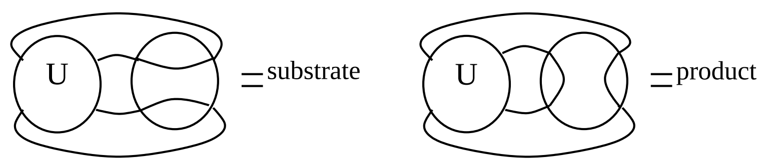


Figure 3.12: Tangle equations modeling *Cre* recombination.

A general system of tangle equations modeling enzyme action is shown in Figure 3.13: $N(U+B) = K_1$ and $N(U+E) = K_2$ where K_1 is the topology of the substrate knot and K_2 is the topology of the product knot. It is hard to identify the DNA configuration bound by protein before removing the protein. Solving tangle equations determines possible topological configurations of DNA bound by protein. The above tangle equations were analyzed in several cases [12, 11, 19, 5, 51]. Isabel K. Darcy solved tangle equations when K_1 and K_2 are 4-plats and B and E are rational knots [12, 11]. She determined U and E in terms of B.

There are software packages which model enzyme actions using tangles and solve tangle equations [15, 14, 44]. TopoICE (Topological Interactive Construction

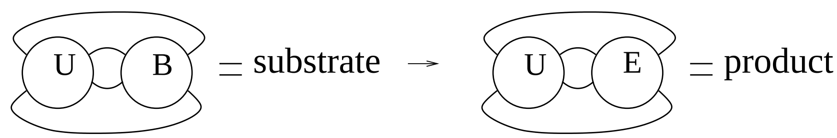


Figure 3.13: System of tangle equations modeling enzyme action.

Engine)-X [15] models topoisomerase reaction and TopoICE-R [14] and TangleSolve [44] model recombination. TopoICE-X and -R are part of KnotPlot.

CHAPTER 4
CALCULATING KNOT DISTANCES

In [48], the Jones polynomial is used to determine in some cases if a knot can be unknotted by switching a positive crossing to the negative. Since the unknotting number is a special case of the knot distance, this motivates the use of the Jones polynomial to improve lower bounds of signed knot distances and thus lower bounds of knot distances. Combined with the Jones polynomial, the signature of a knot is used to improve lower bounds of signed knot distances. Starting with mathematical definitions used in this chapter, mathematical theories and computational algorithms will be discussed.

4.1 Preliminaries

Definition 4.1. A *crossing change* at a crossing is the change of overstrand to understrand or understrand to overstrand.

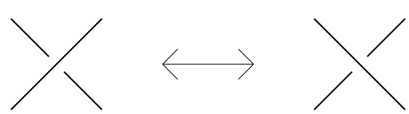


Figure 4.1: Crossing change.

Definition 4.2. [37] Let K_1 and K_2 be knots. Then the *knot distance* between K_1 and K_2, $d(K_1, K_2)$ is defined as the minimum number of crossing changes required to convert K_1 to K_2 where the minimum is taken over all diagrams.

Note that the knot distance satisfies all the properties of a metric.

Definition 4.3. Let K be a knot. Then the *unknotting number* of K is $u(K) = d(K, 0_1)$ where 0_1 is the unknot.

The four link diagrams in Figure 4.2 are identical except at a crossing. The last two diagrams are vertical and horizontal smoothings of the crossing. Note that horizontal smoothing does not preserve the orientation of the original link diagram.

(a) L_+ (b) L_- (c) L_v (d) L_h

Figure 4.2: Partial link diagrams;(a) positive crossing. (b) negative crossing. (c) vertical smoothing. (d) horizontal smoothing.

Now, we define signed crossing changes and signed knot distances.

Definition 4.4. [50]

1. $L_+ \to L_-$ is called a $+-$ crossing change.

2. $L_- \to L_+$ is called a $-+$ crossing change.

Definition 4.5. [50]

1. $d_+(K_1, K_2)$ is the minimum number of $+-$ crossing changes needed to convert K_1 into K_2 where $-+$ crossing changes are allowed, but not counted. There is no restriction on the number of $-+$ crossing changes.

2. $d_-(K_1, K_2)$ is the minimum number of $-+$ crossing changes needed to convert K_1 into K_2 where $+-$ crossing changes are allowed, but not counted. There is no restriction on the number of $+-$ crossing changes.

3. $d_{++}(K_1, K_2)$ is the minimum number of $+-$ crossing changes needed to convert K_1 into K_2 where only $+-$ crossing changes are allowed.

4. $d_{--}(K_1, K_2)$ is the minimum number of $-+$ crossing changes needed to convert K_1 into K_2 where only $-+$ crossing changes are allowed.

Remark. Note that

1. $d_{++}(K_1, K_2) \geq 1$ and $d_{--}(K_1, K_2) \geq 1$ for $K_1 \neq K_2$.

2. If $d_{++}(K_1, K_2) < \infty$, then $d_-(K_1, K_2) = 0$. Similarly, If $d_{--}(K_1, K_2) < \infty$, then $d_+(K_1, K_2) = 0$.

3. $d_+(K_1, K_2) \leq d(K_1, K_2) \leq d_{++}(K_1, K_2)$.

4. $d_-(K_1, K_2) \leq d(K_1, K_2) \leq d_{--}(K_1, K_2)$.

5. $d_+(K_1, K_2) = d_-(K_2, K_1)$.

6. $d_{++}(K_1, K_2) = d_{--}(K_2, K_1)$.

7. None of these distances (d_+, d_-, d_{++} and d_{--}) define a metric on knot/link type. However, all of these distances satisfy the triangle inequality.

By the above properties of signed knot distances, $d(K_1, K_2) \geq Max[d_+(K_1, K_2), d_-(K_1, K_2)]$. So d_+ and d_- can be used to improve lower bounds of knot distances. d_+ has been tabulated for rational knots and composite of rational knots by I.Darcy [10].

Definition 4.6. [43] The *signature* of a knot K, $\sigma(K)$ is defined as the signature of $V + V^T$ where V is the Seifert matrix of K.

Definition 4.7. [31, 32] The *Jones polynomial* of a knot K, $V_K(t)$ is a Laurent polynomial of a variable t which satisfies the following:

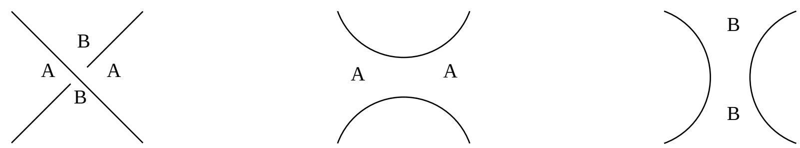
1. $V_{0_1}(t) = 1$ where 0_1 is the unknot.

2. $t^{-1}V_{L_+}(t) - tV_{L_-}(t) + (t^{-\frac{1}{2}} - t^{\frac{1}{2}})V_{L_v}(t) = 0$.

Definition 4.8. [1] The *writhe* of a link diagram D, $w(D)$ is the sum of the signs of the crossings of the link diagram D.

Four regions near a crossing are marked with an A or a B by the following rule: rotate the overstrand counterclockwise until it overlaps the understrand. Then two regions that the overstrand passed over are marked with an A. The other two regions are labeled with a B. See Figure 4.3 (a).

Definition 4.9. [1] A *type A* (respectively B) *splitting* is obtained by connecting two regions marked with an A (respectively B). See Figure 4.3 (b) and (c).



(a) marked region (b) type A splitting (c) type B splitting

Figure 4.3: Splitting.

Definition 4.10. [1] A *state* is a choice of type A or B splitting for all crossings of a diagram.

Definition 4.11. [1] The *bracket polynomial* $< D >$ of a link diagram D is a Laurent polynomial in a variable A obtained by

$$\sum_{S} A^{\sharp A - \sharp B}(-A^2 - A^{-2})^{|S|-1}.$$

where $\sharp A$ and $\sharp B$ denote the number of type A and type B splittings respectively in a state S and $|S|$ is the number of (disjoint) circles obtained after the choice of splittings in the state S.

The bracket polynomial satisfies the following four relations:

1. $< 0_1 >= 1$ where 0_1 is the unknot.

2. $< \times > = A <)(> + A^{-1} < \asymp >$

3. $< \times > = A < \asymp > + A^{-1} <)(>$

4. $< L \cup 0_1 >= (-A^2 - A^{-2}) < L >$

The third relation is identical with the second if the diagrams are rotated by $90°$. We state it for convenience of calculation.

Remark. [1] The Jones polynomial of a link L can be calculated from the bracket polynomial:

$$V_L(t) = (-t^{-\frac{3}{4}})^{-w(D)} < D > |_{A=t^{-\frac{1}{4}}}$$

where D is a diagram of L, $w(D)$ is the writhe of L and $< D >$ is the bracket polynomial of L.

4.2 Properties of the Jones polynomial

The following are properties of the Jones polynomial. These can be used to check whether a given polynomial could be the Jones polynomial of a knot.

1. $V_K(1) = 1$, [32]

2. $V_K'(1) = 0$, [31]

3. $V_K(e^{\frac{2\pi i}{3}}) = 1$, [31, 32]

4. $18|V^{(3)}(1)$, [22]

5. $36|V^{(3)}(1) + 3V^{(2)}(1)$, [22]

6. $1 - V_K(t) = w_K(t)(1-t)(1-t^3)$ for some Laurent polynomial $w_K(t)$, [31, 32]

7. $V_K(e^{\pm\frac{\pi i}{3}}) = \pm(i\sqrt{3})^d$ for $d = \dim H_1(\widetilde{K}, \mathbb{Z}_3)$ where $\widetilde{K}$ is the double branch cover of K [35, 22].

4.3 Improving lower bounds of knot distances

Theorem 3 in [48] is generalized to the knot distance case.

Theorem 4.1. *Let K_1 and K_2 be 1-component knots. If $d_{++}(K_1, K_2) = 1$, then there exists a knot K' with $V_{K'} = t^{-\widetilde{V}'(1)}\widetilde{V}$ where*

$$\widetilde{V} = V_{K_2} - \frac{V_{K_1} - V_{K_2}}{t-1}$$

where V_K is the Jones polynomial of a knot K. Here, K' is obtained by horizontal smoothing of the positive crossing of K_1 where the crossing change occurs.

Proof. Assume that D_1 is a diagram of K_1 which is converted to a diagram D_2 of K_2 by switching a positive crossing of D_1 to the negative. W.L.O.G, we can assume that D_1 is a diagram of K_1 with $w(D_1) = 0$ since adding or removing a Reidemeister 1 move does not affect the knot type but changes the writhe by ± 1. Consider the bracket polynomial of D_1 and D_2. Since $w(D_1) = 0$,

$$< \times > = < D_1 > = \left[(-t^{-\frac{3}{4}})^{w(D_1)} V_{K_1}\right]_{A = t^{-\frac{1}{4}}} = V_{K_1}.$$

Since $w(D_2) = -2$,

$$\begin{aligned} < \times > = < D_2 > =& [(-t^{-\frac{3}{4}})^{w(D_2)} V_{K_2}]_{t=A^{-4}} \\ =& [(-t^{-\frac{3}{4}})^{-2} V_{K_2}]_{t=A^{-4}} = A^{-6} V_{K_2}. \end{aligned}$$

Using the bracket polynomial relations,

$$\begin{pmatrix} A & A^{-1} \\ A^{-1} & A \end{pmatrix} \begin{pmatrix} <)(> \\ < \asymp > \end{pmatrix} = \begin{pmatrix} V_{K_1} \\ A^{-6} V_{K_2} \end{pmatrix}.$$

$$\begin{pmatrix} <)(> \\ < \asymp > \end{pmatrix} = \frac{1}{A^2 - A^{-2}} \begin{pmatrix} A & -A^{-1} \\ -A^{-1} & A \end{pmatrix} \begin{pmatrix} V_{K_1} \\ A^{-6} V_{K_2} \end{pmatrix}.$$

Figure 4.4 shows a partial diagram of D_1 with orientation. According to the orientation of the diagram, the parity of B can be $+1$ or ∞. If the parity of B is $+1$, then D_1 is a two component link diagram. Thus the parity of B must be ∞. Since the parity of B is ∞, horizontal smoothing of the positive crossing in D_1 where the crossing change takes place results in a knot while vertical smoothing results in a link. Let K be the knot obtained by horizontal smoothing of the positive crossing in D_1 where the crossing change takes place. Moreover, the orientation of K' should be given as in Figure 4.5. This results in reversing orientation of only one of the two strings of B if B is considered as a tangle. Note that $w(B) = P - N + R = -1$ with the original orientation (Figure 4.4) where P and N are the number of positive and negative crossing respectively between two strings and R is the number of self-crossing in B. Reversing the orientation of only one string reverses the sign of crossings between two strings but the sign of self-crossing remains the same. Thus $w(K') = N - P + R = 2R + 1$.

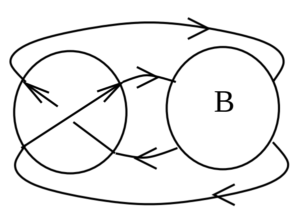


Figure 4.4: A partial diagram of D_1 with orientation.

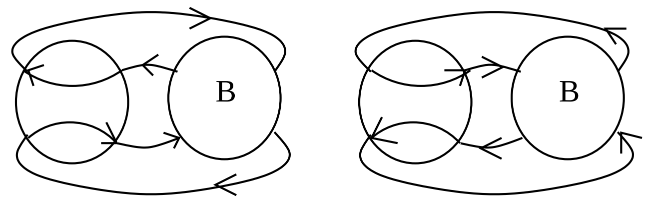


Figure 4.5: Possible partial diagrams of K' with orientation.

$$
\begin{aligned}
< K' > =& \frac{1}{A^2 - A^{-2}}(-A^{-1}V_{K_1} + A^{-5}V_{K_2}) \\
=& \frac{-A^{-3}V_{K_1} + A^{-7}V_{K_2}}{1 - A^{-4}} \\
=& -A^{-3}\left(\frac{V_{K_1} - A^{-4}V_{K_2}}{1 - A^{-4}}\right) \\
=& -A^{-3}\left(\frac{V_{K_1} - tV_{K_2}}{1 - t}\right) \\
=& -A^{-3}\left(V_{K_2} - \frac{V_{K_1} - V_{K_2}}{t - 1}\right) = -t^{\frac{3}{4}}\widetilde{V}.
\end{aligned}
$$

Thus, $\widetilde{V} = V_{K_2} - \dfrac{V_{K_1} - V_{K_2}}{t - 1}$ and

$$
\begin{aligned}
V_{K'} =& (-t^{-\frac{3}{4}})^{-w(K')} < K' > \\
=& (-t^{-\frac{3}{4}})^{-w(K')}(-t^{\frac{3}{4}})\widetilde{V} \\
=& (-1)^{-w(K')+1}t^{\frac{3}{4}(w(K')+1)}\widetilde{V} \\
=& t^{\frac{3}{4}(w(K')+1)}\widetilde{V} \text{ since } w(K') \text{ is odd} \\
=& t^{\alpha}\widetilde{V} \qquad \text{where} \quad \alpha = \frac{3}{4}(w(K') + 1).
\end{aligned}
$$

Then,

$$\widetilde{V} = t^{-\alpha} V_{K'}.$$

Taking the derivative on both sides and evaluating at 1,

$$\widetilde{V}' = -\alpha t^{-\alpha-1} V_{K'} + t^{-\alpha} V'_{K'}.$$

$$\text{Thus} \quad \widetilde{V}'(1) = -\alpha \quad \text{since} \quad V_{K'}(1) = 1 \quad \text{and} \quad V'_{K'}(1) = 0.$$

Hence, $V_{K'} = t^{-\widetilde{V}'(1)} \widetilde{V}$.

□

Theorem 4.2. *Let K_1 and K_2 be 1-component knots. If $d_{--}(K_1, K_2) = 1$, then there exists a knot K' with $V_{K'} = t^{-\widetilde{V}'(1)} \widetilde{V}$ where*

$$\widetilde{V} = V_{K_1} - \frac{V_{K_2} - V_{K_1}}{t - 1}.$$

where V_K is the Jones polynomial of a knot K. Here, K' is obtained by horizontal smoothing of the negative crossing of K_1 where the crossing change occurs.

Proof. If we switch K_1 and K_2 and apply theorem 4.1, we get the result. □

4.4 Properties of signature

Suppose the Jones polynomial improves only one of $d_{++}(K_1, K_2)$ and $d_{--}(K_1, K_2)$. Assume that only $d_{++}(K_1, K_2)$ is changed from one to two based on the Jones polynomial. That is, $d_{++}(K_1, K_2) \geq 2$. Then we addressed a possibility that other knot invariants can be used to determine whether $d_{--}(K_1, K_2) \geq 2$ which concludes that $d(K_1, K_2) \geq 2$. Here, we employ the signature of knots for that purpose. See definition 4.6.

1. $\sigma(K)$ is an even number.

2. $\sigma(L_+) \leq \sigma(L_-) \leq \sigma(L_+) + 2$. [38, 25]

   In other words, a $+-$ crossing change can increase the signature by at most 2.

3. $|\sigma(K_1) - \sigma(K_2)| \leq 2d(K_1, K_2)$. [37]

4. $-2d_{++}(K_1, K_2) \leq \sigma(K_1) - \sigma(K_2) \leq 0$. [10]

5. $-2d_{+}(K_1, K_2) \leq \sigma(K_1) - \sigma(K_2)$. [10]

6. $0 \leq \sigma(K_1) - \sigma(K_2) \leq 2d_{--}(K_1, K_2)$.[10]

7. $\sigma(K_1) - \sigma(K_2) \leq 2d_{--}(K_1, K_2)$. [10]

Property 2 can be restated as follows in lemmas 4.3 and 4.4.

Lemma 4.3. *[38, 25] Let K_1 and K_2 be knots.*

If $\sigma(K_2) > \sigma(K_1) + 2$, then $d_{++}(K_1, K_2) \geq 2$.

Lemma 4.4. *[38, 25] Let K_1 and K_2 be knots.*

If $\sigma(K_1) - 2 > \sigma(K_2)$, then $d_{--}(K_1, K_2) \geq 2$.

4.5 Methodology

We summarize how theorems 4.1 and 4.2 and lemmas 4.3 and 4.4 are used.

1. Generate a list of pairs of knots (K_1, K_2) whose distance has lower bound 1 in the knot distance table.

2. Calculate $\widetilde{V}$ and $V_{K'}$ in theorem 4.1.

3. Check whether $V_{K'}$ can be the Jones polynomial of a knot using properties of the Jones polynomial 1-7 in section 4.2. If $V_{K'}$ violates at least one of those properties, $V_{K'}$ is not the Jones polynomial of a knot. So $d_{++}(K_1, K_2) \geq 2$.

4. Determine $d_{++}(K_1, K_2) \geq 2$ by checking the signature condition in lemma 4.3.

5. Repeat the above processes to check whether $d_{--}(K_1, K_2) \geq 2$ using theorem 4.2 and lemma 4.4.

6. If $d_{++}(K_1, K_2) \geq 2$ and $d_{--}(K_1, K_2) \geq 2$, then $d(K_1, K_2) \geq 2$

In the next section, we describe how steps 2 and 3 are implemented computationally. Other steps are obvious and easy to handle.

4.6 Algorithm

First, we calculate the Jones polynomial of rational knots, composites of rational knots up to 13 crossings and 8 and 9 crossing nonrational prime knots. Then, given two Jones polynomials whose corresponding knots have lower bound one for the knot distance between them, polynomials in theorems 4.1 and 4.2 are calculated.

These polynomials are examined whether they are not the Jones polynomial of a knot by checking properties of the Jones polynomial.

Bruce Ewing and Kenneth C. Millett developed a computational algorithm for calculating the HOMFLYPT polynomial [21]. The HOMFLYPT polynomial is named after its co-discoverers: Hoste, Ocneanu, Millett, Freyd, Lickorish, and Yetter [24] plus Przytycki and Traczyk [41]. The HOMFLYPT polynomial, $P_L(l, m)$ is a Laurent polynomial with two variables l and m and integer coefficients defined by the following relations:

1. $P_{0_1}(l, m) = 1$ where 0_1 is the unknot.
2. $lP_{L_+}(l, m) + l^{-1}P_{L_-}(l, m) + mP_{L_0}(l, m) = 0$.

Since the Jones polynomial is a special case of the HOMFLYPT polynomial, it can be obtained from the HOMFLYPT polynomial by substituting l and m with it^{-1} and $i(t^{-\frac{1}{2}} - t^{\frac{1}{2}})$, respectively:

$$V_K(t) = P_L(l = it^{-1}, m = i(t^{-\frac{1}{2}} - t^{\frac{1}{2}})).$$

For the computational algorithm for HOMFLYPT polynomial calculation, Bruce Ewing and Kenneth C. Millett invented a knot code referred to as the *Ewing-Millett code* or *EM code*. The following is an EM code for the trefoil knot:

Trefoil

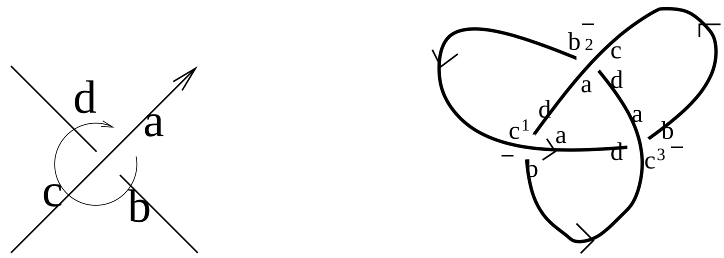
1-3d3c2b2a

2-1d1c3b3a

3-2d2c1b1a

The code consists of the knot name in the first row and information about each crossing in the following rows. The first row does not affect the calculations, and thus it can be any format. The crossing information for a given crossing is represented by listing the number assigned to the crossing, the sign of the crossing and four sets of two symbols which indicate where four endpoints of two strands at the crossing connect to. Here, the four endpoints represent where a 3-ball around the crossing meets the knot. The four endpoints at each crossing are marked with a,b,c and d as in Figure 4.6 (a). Given an orientation of the knot, the outgoing direction on the overstrand at a crossing is labeled as 'a' and then the other endpoints are marked with 'b','c' and 'd' in the clockwise direction. Figure 4.6 (b) shows a labeling of the trefoil knot with the number assigned to the crossing, its sign and the letters at each crossing. In figure 4.6, the crossing number '1' is negative and the 'a' of the crossing '1' is connected to the 'd' of the crossing '3' and the 'b' of the crossing '1' is connected to the 'c' of the crossing '3' etc. This connecting information is encoded in the second row of the above EM code for trefoil knot: 1-3d3c2b2a.

The Jones polynomials are calculated by using B. Ewing and K. Millett's algorithm for calculating the HOMFLYPT polynomial. For that purpose, we developed an algorithm generating EM codes for rational knots. Here, the 4-plat notation is used for rational knots. A rational knot $N(\frac{a}{-b})$ is equal to the 4-plat $S(a,b)$ or



(a) (b)

Figure 4.6: EM code;(a) Letter assignment. (b)Diagram for coding the trefoil knot.

$< x_1, \cdots, x_n >$ where $\dfrac{a}{b} = x_1 + \dfrac{1}{\frac{1}{x_2+\cdots+\frac{1}{x_n}}}$. Using the Euclidean algorithm, every rational number can be expanded into a vector of integer entries. If either a or b is even, $\dfrac{a}{b}$ can always be expanded into a vector whose entries are all even by means of a modified division algorithm. For example, an expansion of the rational number $\dfrac{3}{2}$ is $2 + \dfrac{1}{-2}$ which corresponds to the 4-plat $< 2, -2 >$. If both a and b are odd, by theorem 2.1, $N(\dfrac{a}{b}) = N(\dfrac{a}{a+b})$ since $a + b = b$ mod a. Then $\dfrac{a}{a+b}$ can be expanded as a vector whose entries are all even. Representing a rational knot with all even integers has the advantage that keeping track of the orientation of the knot diagram is easy. An orientation is given and crossing numbers are assigned to the knot diagram drawn from the corresponding vector of even integers for a knot. Figure 4.7 shows an example of encoding the knot $S(11, 4)$. The partial EM code in Figure 4.7 includes sufficient information to draw the knot but empty places are also filled out for efficiency of running the Ewing Millett program. The programming code for generating EM codes for rational knots is provided in Appendix B.

For the EM codes for nonrational 8 and 9 crossing knots, we used Knotscape by Jim Hoste and Morwen Thistlethwaite [29]. In Knotscape, an EM code of a knot is obtained from a diagram of the knot.

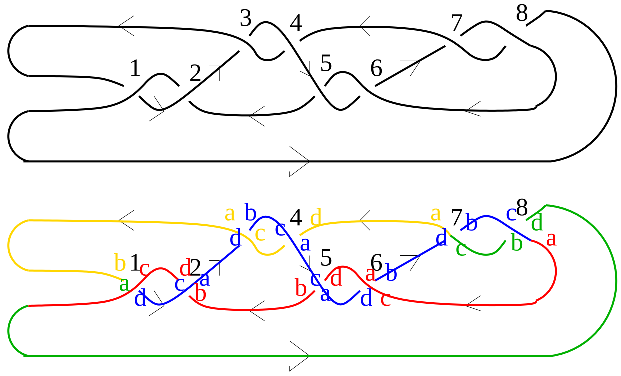
To obtain the Jones polynomials using EM code, Ewing-Millett's code for calculating the HOMFLYPT polynomial is modified by replacing the variables l and m with it and $i(t^{-\frac{1}{2}} - t^{\frac{1}{2}})$, respectively [1]. Knowing that the HOMFLYPT polynomial of a knot has all even powers, l^2 is replaced with $-t^2$ and m^2 is replaced with $-(t^{-1} - 2 + t)$ to make the calculation easier.

With the list of Jones polynomials and the list of pairs of knots whose knot distances have lower bound one, the next step is to calculate the corresponding polynomials in theorem 4.1 and theorem 4.2 and then check whether those polynomials violate one of the Jones polynomial's properties given in section 4.2.

The signatures for rational knots and composites of rational knots up to 13 crossing and nonrational prime 8 crossing knots [10] are used to check the conditions in lemma 4.3 and lemma 4.4. A code in MATLAB is used to detect pairs of knots whose distances are improved by the Jones polynomial and the signature. MATLAB is useful to deal with data import and polynomial manipulation.

4.7 New knot distance table

From more than 50,000 pairs of knots whose knot distances have lower bound 1, we could detect 6,457 pairs which show improvement on their lower bound from one to two based only on the Jones polynomial. The signature did not discover any changes on signed knot distances which the Jones polynomial did not improve. Among those improved, some unknotting numbers are already known. For example, the unknotting number of 8_{18} was improved from one to two using the Jones polynomial and it is already known as 2 [48]. The new knot distance table confirms that the unknotting number of 8_{18} is 2. I modified Isabel Darcy's programs to generate a new knot distance table. In the new knot distance table given in Appendix C, the updated distances are colored red.



$1+$ $8d$ $2c$

$2+$ $3d$ $1c$

$3-$ $1b$ $4c$

$4-$ $5c$ $3c$

$5-$ $6d$ $2b$

$6-$ $5d$ $7d$

$7-$ $4d$ $8c$

$8-$ $6c$ $7c$

Figure 4.7: Coding $S(11,4) =< 2,2,-2,2 >$.

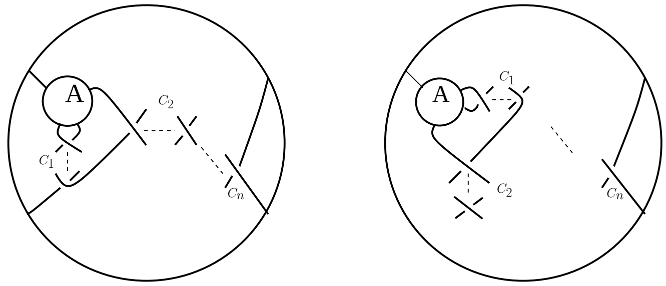
CHAPTER 5
SOLVING TANGLE EQUATIONS

In many biological applications, the knots/links involved are rational knots/links. However, Montesinos knots/links are also observed in computer simulations and they are proposed to be possible configurations of product DNA in some tangle analysis modeling recombination [4]. In this chapter, the system of unoriented tangle equations $N(U + \frac{0}{1}) = K_1$ and $N(U + \frac{x}{y}) = K_2$ is solved for U assuming K_1 and K_2 are either Montesinos links or rational links, $\frac{x}{y}$ is a rational tangle and U is a generalized Montesinos tangle. The system of tangle equations where K_1 and K_2 are rational links is solved in [12]. Thus we solve the system of tangle equations when at least one of K_1 or K_2 is a Montesinos link in this chapter.

5.1 Preliminaries

Definition 5.1. The *circle product*, $A \circ (c_1, \cdots, c_n)$ of two tangles A and $(c_1, \cdots, c_n)$ is obtained by starting with c_1 vertical (horizontal) half twists of SW and SE (NE and SE) endpoints of A and alternating between horizontal (vertical) half twists and vertical (horizontal) half twists when n is even (odd) (Figure 5.1).

Definition 5.2. [12] A *generalized Montesinos tangle* or *generalized M-tangle* is a tangle of the form $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_m)$ where $\frac{a_i}{b_i}$'s are rational tangles, $\frac{a_i}{b_i} \neq \frac{1}{0}$ for $1 \leq i \leq n$ and h_j's are integers for $1 \leq j \leq m$. An *M-tangle* is a tangle of the form $\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}$ where $\frac{a_i}{b_i}$'s are rational tangles and $\frac{a_i}{b_i} \neq \frac{1}{0}$ for $1 \leq i \leq n$.



n:even n:odd

Figure 5.1: Circle product $A \circ (c_1, \cdots, c_n)$.

Lemma 5.1. *[12]*

1. *A generalized M-tangle is rational if all but at most one of the* $\dfrac{a_i}{b_i}$*'s are integral.*

2. *The sum of two rational tangles is rational if and only if one of the tangles is integral. In this case,* $\dfrac{a}{b} + x = x + \dfrac{a}{b} = \dfrac{a+bx}{b}$.

3. $\dfrac{a}{b} + \dfrac{c}{d} = \dfrac{a}{b} - x + x + \dfrac{c}{d} = \dfrac{a-bx}{b} + \dfrac{c+dx}{d}$.

As you can see in Figure 5.1, circle product manipulates 4 strings connected from NW, NE, SW and SE endpoints of A in $S^2 \times I$. Actually, NW endpoint of A is fixed and other three endpoints are manipulated while we do circle product. Thus $(c_1, \cdots, c_n)$ in $A \circ (c_1, \cdots, c_n)$ can be considered as a 3-string tangle. A 3-string tangle is a 3-ball with 3 strings embedded in the ball and 6 endpoints are fixed on the boundary of the ball (Figure 5.3 (b)). The 3-string tangles used in the circle product will be related to 3-braids (See definition 5.2) which will be used to classify generalized M-tangles in theorem 5.15.

Note that even if two rational tangles A and B are equivalent, it is possible that $U \circ A \neq U \circ B$ for a tangle U. For example, $(3, -3) = (-2, -1, -2)$ but $(2) \circ (3, -3) \neq (2) \circ (-2, -1, -2)$ since $(2) \circ (3, -3) = (2, 3, -3) = \frac{-19}{7}$ but $(2) \circ (-2, -1, -2) = (0, -1, -2) = -2$. Note that $(3, -3)$ and $(-2, -1, -2)$ are equivalent as 2-string tangles since they have the same continued fraction $\dfrac{-8}{3}$ but they are not equivalent as 3-string tangles in $(2) \circ (3, -3)$ and $(2) \circ (-2, -1, -2)$ since $(2) \circ (3, -3)$ cannot be deformed into $(2) \circ (-2, -1, -2)$ without moving NW, NE, SW and SE endpoints of $(2) \circ (3, -3)$.

Now, we define a 3-braid and discuss circle product in terms of a sum between a 2 string tangle and a 3-braid.

Definition 5.3. [7, 8] A *3-braid* is a set of 3 strings which are attached to vertical bars at the left and at the right as in Figure 5.3 (a). Each string always heads to the right as we move along the string from the left vertical bar to the right vertical bar. We denote a 3-braid by $T(a_1, \cdots, a_n)$ as in Figure 5.2 where $a_i \in Z$ [7, 8]:

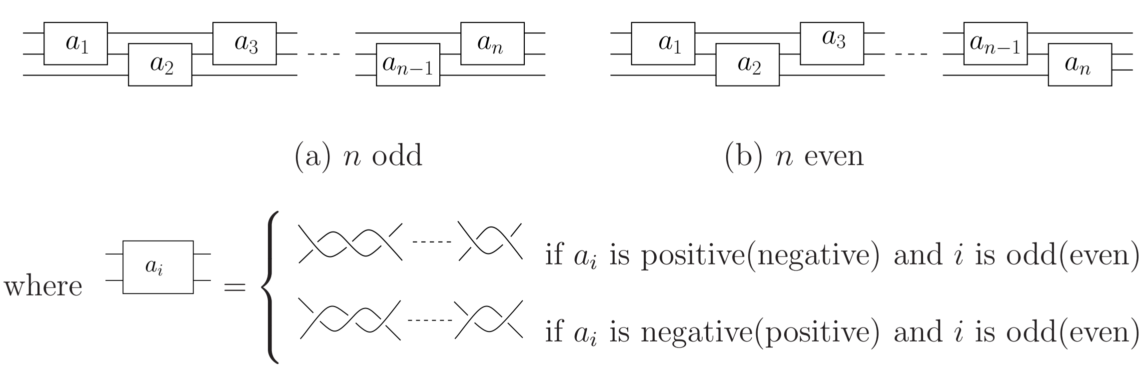
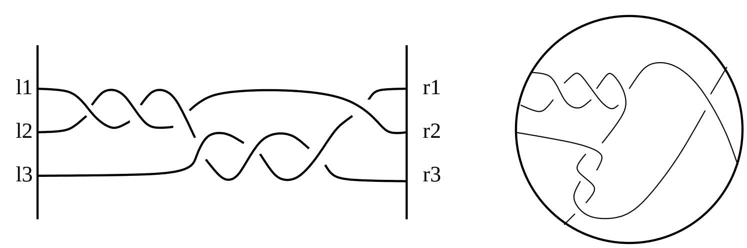


Figure 5.2: $T(a_1, \cdots, a_n)$.

There is the standard notation for a 3-braid defined as follows [2]: If the first string crosses over (under) the second, then the crossing is called σ_1 (σ_1^{-1}). If the second string crosses over (under) the third, then the crossing is called σ_2 (σ_2^{-1}). Thus standard notation of the 3-braid in Figure 5.3 (a) is $\sigma_1^3\sigma_2^{-3}\sigma_1^1$. For more details for braid notation, see [2]. Note that a 3-braid notation $T(a_1, \cdots, a_n)$ is equal to $\sigma_1^{a_1}\sigma_2^{-a_2}\cdots\sigma_1^{a_n}$ if n is odd and $\sigma_1^{a_1}\sigma_2^{-a_2}\cdots\sigma_2^{-a_n}$ if n is even. In this chapter, we use the notation $T(a_1, \cdots, a_n)$ instead of the standard notation to relate a 3-braid to a 3-string tangle in circle product.

A 3-braid can be considered as a special case of a 3-string tangle as in Figure 5.3. The braid sum, $A +_b B$ of two 3-braids A and B is defined by connecting r_1, r_2, r_3 of A to l_1, l_2, l_3 of B, respectively [7, 8].

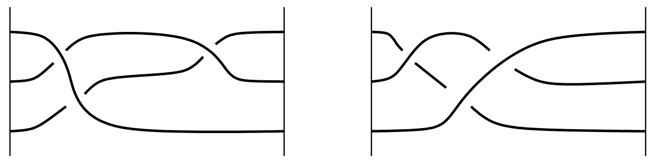


(a) $T(3,3,1)$ (b)

Figure 5.3: Converting a 3-braid into a 3-string tangle;(a) A 3-braid and (b) The corresponding 3-string tangle.

Every braid has a standard diagram as in the following theorem.

Theorem 5.2. *[7] If B is a 3-braid, then $B \cong T(g_1, \cdots, g_k) +_b tE$ where g_i's have* *the same sign and E is shown in Figure 5.4 (a).*



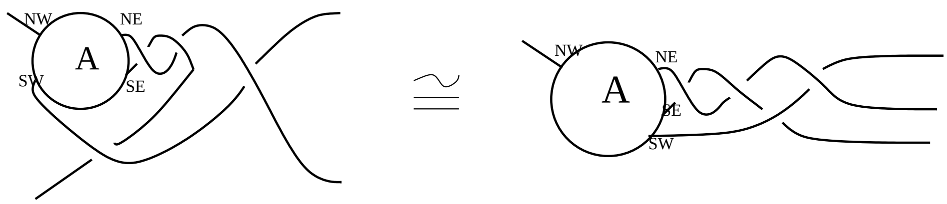
(a)$E = T(1, -1, 1)$ (b)$-E = T(-1, 1, -1)$

Figure 5.4: Braids;(a) E and (b) $-E$.

Since $(g_1, \cdots, g_k)$ in $A \circ (g_1, \cdots, g_k)$ can be considered as a 3-braid by placing vertical twists horizontally and moving the SW endpoints of A and $A \circ (g_1, \cdots, g_k)$ to the east sides of A and $A \circ (g_1, \cdots, g_k)$ respectively as in Figure 5.5, circle product can be related to braid sum as follows: denote $A \circ (g_1, \cdots, g_k) = A +_{tb} T(g_1, \cdots, g_k)$ where A is a 2-string tangle, $(g_1, \cdots, g_k)$ is a 3-string tangle and $T(g_1, \cdots, g_k)$ is the corresponding 3-braid. The operation $'+'_{tb}$ represents the sum between a 2-string tangle A and a 3-braid $T(g_1, \cdots, g_k)$ by connecting NE, SE and SW endpoints of A to l_1, l_2 and l_3 of $T(g_1, \cdots, g_k)$.

The vertical sum, $A * B$ of tangles A and B is defined by connecting SW and SE endpoints of A with NW and NE endpoints of B, respectively [33].

Theorem 5.3. *[7]*



(a)$A \circ (2,1,1)$ (b)$A +_{tb} T(2,1,1)$

Figure 5.5: Circle product as the sum between a 2 string tangle and a 3-braid; (a) Circle product and (b) The corresponding sum.

Let $T(h_1, \cdots, h_m) = T(g_1, \cdots, g_k) +_b tE$. *Then*

$$(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_m) = \begin{cases} (\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (g_1, \cdots, g_k) & \text{if } t \text{ is even} \\ (\frac{-b_1}{a_1} * \cdots * \frac{-b_n}{a_n}) \circ (-g_1, \cdots, -g_k, 0) & \text{if } t \text{ is odd} \end{cases}$$

Proof. $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_m) = (\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) +_{tb} T(h_1, \cdots, h_m)$
$= (\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) +_{tb} [T(g_1, \cdots, g_k) +_b tE] = [(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) +_{tb} T(g_1, \cdots, g_k)] +_{tb} tE$
$= [(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (g_1, \cdots, g_k)] \circ (1, -1, 1) \circ (1, -1, 1) \circ \cdots \circ (1, -1, 1)$.
Let $A = (\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (g_1, \cdots, g_k)$. Let $r_A(T)$ be the rotation of a tangle A by 180° about the line from the NW and the SE of A where $T = A \circ (1, -1, 1) \circ (1, -1, 1) \circ \cdots \circ (1, -1, 1)$ and $r(T)$ be the rotation of a tangle T by 180° about the line from the NW and the SE of T. Note that $r_A(A \circ (1, -1, 1)) = r(A)$ and $r_A^2(A \circ (1, -1, 1) \circ (1, -1, 1)) = A$. Thus by induction, $A \circ (1, -1, 1) \circ (1, -1, 1) \circ \cdots \circ (1, -1, 1)$ is deformed into A if t is even and $r(A)$ if t is odd. Note that $r(A) = r((\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (g_1, \cdots, g_k)) = (r(\frac{a_1}{b_1}) * \cdots * r(\frac{a_n}{b_n})) \circ (-g_1, \cdots, -g_k, 0) = (\frac{-b_1}{a_1} * \cdots * \frac{-b_n}{a_n}) \circ (-g_1, \cdots, -g_k, 0)$. □

By lemma 5.1 and theorem 5.3, a generalized M-tangle can be represented as $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_m)$ or $(\frac{-b_1}{a_1} * \cdots * \frac{-b_n}{a_n}) \circ (-h_1, \cdots, -h_m, 0)$ where a_i and

b_i are relatively prime and $0 < a_i < b_i$. Generalized M-tangles will be classified in theorem 5.15.

Definition 5.4. [3, 6, 56] A *Montesinos knot/link* has a projection as shown in Figure 5.6 where e is an integral tangle and $\frac{a_i}{b_i}$ is a rational tangle for $i = 1, \cdots, r$ and $r \geq 3$. Here, we assume a_i and b_i are relatively prime and $0 < a_i < b_i$. This implies that $\frac{a_i}{b_i}$ is neither an integral tangle nor the infinity tangle. The above Montesinos link is written as $N(\frac{a_1}{b_1} + \cdots + \frac{a_r}{b_r} + e)$.

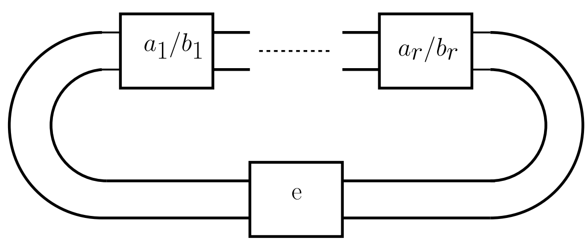


Figure 5.6: Montesinos link.

Note that Montesinos links do not include rational links/knots according to the above definition although generalized M-tangles include rational tangles.

Theorem 5.4. (Classification of Montesinos Links) *[3, 6, 56] Montesinos links with r rational tangles, $r \geq 3$, are classified by the ordered set of fractions $(\frac{a_1}{b_1}, \cdots, \frac{a_r}{b_r})$, up to cyclic permutations and reversal of order, together with the integer e where a_i and b_i are coprime integers such that $0 < a_i < b_i$ for $1 \leq i \leq r$.*

Lemma 5.5. *$N(A + C) = N(C + A)$ where A and C are arbitrary tangles.*

Lemma 5.6. *[12] $N(A \circ (c_1, \cdots, c_n) + B) = N(A + B \circ (c_n, \cdots, c_1))$ if A or B is rational and n is odd.*

Lemma 5.7. *$N(A \circ (c_1, \cdots, c_n) + B) = N(A + B \circ (c_n, \cdots, c_1))$ if n is odd and B is invariant under $180°$ rotations about x and y axis.*

Proof. If $n = 1$, $N(A \circ (c_1) + B) = N(A + B \circ (c_1))$ since B is invariant under $180°$ rotation about x axis. Assume that $N(A \circ (c_1, \cdots, c_n) + B) = N(A + B \circ (c_n, \cdots, c_1))$ when $n = 2k - 1$ for $k \geq 1$. Then $N(A \circ (c_1, \cdots, c_{2k+1}) + B) = N(A \circ (c_1, \cdots, c_{2k}, 0) + B \circ (c_{2k+1})) = N(A \circ (c_1, \cdots, c_{2k-1}) \circ (c_{2k}, 0) + B \circ (c_{2k+1})) = N(A \circ (c_1, \cdots, c_{2k-1}) + B \circ (c_{2k+1}) \circ (c_{2k}, 0)) = N(A \circ (c_1, \cdots, c_{2k-1}) + B \circ (c_{2k+1}, c_{2k}, 0))$. Since B is invariant under $180°$ rotations about x and y axis, so are $B \circ (c_{2k+1})$ and then $B \circ (c_{2k+1}) \circ (c_{2k}, 0)$. Thus by the induction hypothesis, $N(A \circ (c_1, \cdots, c_{2k-1}) + B \circ (c_{2k+1}, c_{2k}, 0)) = N(A + B \circ (c_{2k+1}, c_{2k}, 0) \circ (c_{2k-1}, \cdots, c_1)) = N(A + B \circ (c_{2k+1}, c_{2k}, c_{2k-1}, \cdots, c_1))$. By induction, we have the result. $\square$

Note that a rational tangle is invariant under $180°$ rotations about x and y axis. Thus lemma 5.6 is a special case of lemma 5.7.

Lemma 5.8. *[12] $(d_1, \cdots, d_m) \circ (c_1, \cdots, c_n) = (d_1, \cdots, d_m + c_1, \cdots, c_n)$ if n is odd.*

Definition 5.5. [42] The *Euler bracket function*, $E[x_1, \cdots, x_n]$ equals the sum of the products obtained from the product $1 \cdot x_1 \cdot \cdots \cdot x_n$ by omitting zero or more disjoint pairs of consecutive $x_i x_{i+1}$ from the product.

The Euler bracket function satisfies the following.

Proposition 5.9. *1. If $n = 0$, then $E[x_1, \cdots, x_n] = E[] = 1$.*

2. If $n < 0$, then $E[x_1, \cdots, x_n] = 0$.

3. $E[x_1, \cdots, x_n] = E[x_n, \cdots, x_1]$.

4. For $n \geq 1$,

(a) $E[x_1, \cdots, x_n] = x_1 E[x_2, \cdots, x_n] + E[x_3, \cdots, x_n]$

$= x_n E[x_1, \cdots, x_{n-1}] + E[x_1, \cdots, x_{n-2}]$.

(b) $[x_n, \cdots, x_1] = \dfrac{E[x_1, \cdots, x_n]}{E[x_1, \cdots, x_{n-1}]} = \dfrac{E[x_n, \cdots, x_1]}{E[x_{n-1}, \cdots, x_1]}$
where $[x_n, \cdots, x_1] = x_n + \dfrac{1}{x_{n-1} + \cdots + \frac{1}{x_1}}$.

(c) Let $a = E[x_1, \cdots, x_n]$,$b = E[x_1, \cdots, x_{n-1}]$. If $y = (-1)^{n+1} E[x_2, \cdots, x_{n-1}]$ and $x = (-1)^{n+1} E[x_2, \cdots, x_n]$, then $bx - ay = 1$.

Lemma 5.10. *[42] $[c_1, \cdots, c_n + d_m, \cdots, d_1] = \frac{E[c_1, \cdots, c_n] E[d_1, \cdots, d_{m-1}] + E[c_1, \cdots, c_{n-1}] E[d_1, \cdots, d_m]}{E[c_2, \cdots, c_n] E[d_1, \cdots, d_{m-1}] + E[c_2, \cdots, c_{n-1}] E[d_1, \cdots, d_m]}$.*

Theorem 5.11. *[9, 33] Every rational tangle has a unique canonical form $(x_1, \cdots, x_n)$ where $x_i \in Z - \{0\}$ for $1 \leq i \leq n-1$, all x_i's have the same sign and n is odd.*

Proposition 5.12. *Suppose that $0 < a_i < b_i$ for $1 \leq i \leq n$, h_j's have the same sign for all j, $h_j \neq 0$ for $2 \leq j \leq t-1$ and t is odd. For $n \geq 2$, $(\dfrac{a_1}{b_1} + \cdots + \dfrac{a_n}{b_n}) \circ (h_1, \cdots, h_t) = (\dfrac{c_1}{d_1} + \cdots + \dfrac{c_m}{d_m}) \circ (k_1, \cdots, k_s)$ where $0 < c_i < d_i$ for $1 \leq i \leq m$, k_j's have the same sign for all j, $k_j \neq 0$ for $2 \leq j \leq s-1$ and s is odd iff (a) $n = m$ and $\dfrac{a_i}{b_i} = \dfrac{c_i}{d_i}$ for all i and (b) $t = s$ and $h_j = k_j$ for all j.*

Proof. ($\Rightarrow$) Suppose that $(\dfrac{a_1}{b_1} + \cdots + \dfrac{a_n}{b_n}) \circ (h_1, \cdots, h_t) = (\dfrac{c_1}{d_1} + \cdots + \dfrac{c_m}{d_m}) \circ (k_1, \cdots, k_s)$. Then $(\dfrac{a_1}{b_1} + \cdots + \dfrac{a_n}{b_n}) = (\dfrac{c_1}{d_1} + \cdots + \dfrac{c_m}{d_m}) \circ (k_1, \cdots, k_s) \circ (-h_t, \cdots, -h_1) = (\dfrac{c_1}{d_1} +$

$\cdots + \frac{c_m}{d_m}) \circ (k_1, \cdots, k_s - h_t, \cdots, -h_1)$ by lemma 5.8. We can choose a nonintegral rational tangle $\frac{x}{y}$ such that $\frac{x}{y} \neq \frac{1}{0}$ and $\frac{x}{y} \neq (\frac{c_i}{d_i} + e_1)$ for any i where e_1 is an arbitrary integral tangle. Then $N(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n} + \frac{x}{y}) = N((\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m}) \circ (k_1, \cdots, k_s - h_t, \cdots, -h_1) + \frac{x}{y}) = N(\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m} + \frac{x}{y} \circ (-h_1, \cdots, -h_t + k_s, \cdots, k_1))$ by lemma 5.6. Since $\frac{x}{y} \neq \frac{1}{0}$, $\frac{x}{y} \neq (\frac{c_i}{d_i} + e)$ and $\frac{x}{y}$ is not an integral tangle, by theorem 5.4, $\frac{x}{y} = (\frac{x}{y} \circ (-h_1, \cdots, -h_t + k_s, \cdots, k_1))$. Let $\frac{x}{y} = (l_1, \cdots, l_u) = \frac{E[l_1, \cdots, l_u]}{E[l_1, \cdots, l_{u-1}]}$.
Then $\frac{x}{y} \circ (-h_1, \cdots, -h_t + k_s, \cdots, k_1) = (l_1, \cdots, l_u) \circ (-h_1, \cdots, -h_t + k_s, \cdots, k_1) = (l_1, \cdots, l_u - h_1, \cdots, -h_t + k_s, \cdots, k_1) = [k_1, \cdots, k_s - h_t, \cdots, -h_1 + l_u, \cdots, l_1] =$

$$\frac{E[k_1, \cdots, k_s - h_t, \cdots, -h_1]E[l_1, \cdots, l_{u-1}] + E[k_1, \cdots, k_s - h_t, \cdots, -h_2]E[l_1, \cdots, l_u]}{E[k_2, \cdots, k_s - h_t, \cdots, -h_1]E[l_1, \cdots, l_{u-1}] + E[k_2, \cdots, k_s - h_t, \cdots, -h_2]E[l_1, \cdots, l_u]}$$
$$= \frac{E[k_1, \cdots, k_s - h_t, \cdots, -h_1]y + E[k_1, \cdots, k_s - h_t, \cdots, -h_2]x}{E[k_2, \cdots, k_s - h_t, \cdots, -h_1]y + E[k_2, \cdots, k_s - h_t, \cdots, -h_2]x} = \frac{x}{y}.$$

Let $E[k_1, \cdots, k_s - h_t, \cdots, -h_1] = a, E[k_1, \cdots, k_s - h_t, \cdots, -h_2] = b, E[k_2, \cdots, k_s - h_t, \cdots, -h_1] = a'$ and $E[k_2, \cdots, k_s - h_t, \cdots, -h_2] = b'$. Then $\frac{ay + bx}{a'y + b'x} = \frac{x}{y}$ where $(a, b) = 1$ and $(x, y) = 1$. This implies that $ay + bx = kx$ for some integer k. Then $ay = (k-b)x$. Since $(x, y) = 1$, $x|a$. This is true for any x such that $\frac{x}{y} \neq \frac{1}{0}$ and $\frac{x}{y} \neq (\frac{c_i}{d_i} + e_1)$ for any i and any e_1. Thus $a = E[k_1, \cdots, k_s - h_t, \cdots, -h_1] = 0$ and $(-h_1, \cdots, -h_t + k_s, \cdots, k_1) = \frac{E[k_1, \cdots, k_s - h_t, \cdots, -h_1]}{E[k_2, \cdots, k_s - h_t, \cdots, -h_1]} = 0$. Since $(-h_1, \cdots, -h_t + k_s, \cdots, k_1) = [k_1, \cdots, k_s - h_t, \cdots, -h_1] =$

$$\frac{E[k_1, \cdots, k_s]E[-h_1, \cdots, -h_{t-1}] + E[k_1, \cdots, k_{s-1}]E[-h_1, \cdots, -h_t]}{E[k_2, \cdots, k_s]E[-h_1, \cdots, -h_{t-1}] + E[k_2, \cdots, k_{s-1}]E[-h_1, \cdots, -h_t]}$$
$$= \frac{E[k_1, \cdots, k_s]E[h_1, \cdots, h_{t-1}] - E[k_1, \cdots, k_{s-1}]E[h_1, \cdots, h_t]}{E[k_2, \cdots, k_s]E[h_1, \cdots, h_{t-1}] - E[k_2, \cdots, k_{s-1}]E[h_1, \cdots, h_t]} = \frac{0}{1}$$ by lemma 5.10,

$E[k_1, \cdots, k_s]E[h_1, \cdots, h_{t-1}] - E[k_1, \cdots, k_{s-1}]E[h_1, \cdots, h_t] = 0$. Thus $\frac{E[k_1, \cdots, k_s]}{E[k_1, \cdots, k_{s-1}]} = \frac{E[h_1, \cdots, h_t]}{E[h_1, \cdots, h_{t-1}]}$ which implies $(k_1, \cdots, k_s) = (h_1, \cdots, h_t)$ as 2-string tangles.

(1) Assume that $h_1 \neq 0$. If $k_1 \neq 0$, then $t = s$ and $h_j = k_j$ for all j by the-

orem 5.11 since t and s are odd, all h_j's have the same sign, all k_j's have the same sign, $h_j \in Z - \{0\}$ for $1 \leq j \leq t-1$ and $k_j \in Z - \{0\}$ for $1 \leq j \leq s-1$. If $k_1 = 0$, then $(h_1, \cdots, h_t) = (0, k_2, k_3, \cdots, k_s) = (k_3, \cdots, k_s)$. Thus $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t) = (\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m}) \circ (0, k_2, h_1, \cdots, h_t)$ which implies that $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) = (\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m}) \circ (0, k_2)$. Then $N(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n} + \frac{x}{y}) = N((\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m}) \circ (0, k_2) + \frac{x}{y}) = N(\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m} + \frac{x}{y} \circ (0, k_2))$ for a nonintegral rational tangle $\frac{x}{y}$ such that $\frac{x}{y} \neq \frac{1}{0}$ and $\frac{x}{y} \neq (\frac{c_i}{d_i} +$ an integral tangle) for any i. By theorem 5.4, $\frac{x}{y} = (\frac{x}{y} \circ (0, k_2)) = \frac{x}{k_2 x + y}$. If $\frac{x}{k_2 x + y} = \frac{-x}{-y}$, then $k_2 x + y = -y$ which implies $k_2 x = -2y$. Since $(x, y) = 1$, $x = \pm 1, \pm 2$. If we choose x such that $x \geq 3$, then $k_2 = 0$. This contradicts the hypothesis.

(2) Assume that $h_1 = 0$. If $k_1 \neq 0$, then by the similar argument as in (1), $h_2 = 0$ which contradicts the hypothesis. If $k_1 = 0$, then $(h_3, \cdots, h_t) = (k_3, \cdots, k_s)$ as 2-string tangles. By theorem 5.11, $t = s$ and $h_j = k_j$ for $3 \leq j \leq t$. Then $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (0, h_2, h_3, \cdots, h_t) = (\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m}) \circ (0, k_2, h_3, \cdots, h_t)$ which implies that $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) = (\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m}) \circ (0, k_2 - h_2)$. By the same argument as in (1), $k_2 = h_2$.

Since $s = t$ and $h_j = k_j$ for all j, $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t) = (\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m}) \circ (k_1, \cdots, k_s)$ implies that $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) = (\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m})$. Now, choose another nonintegral rational tangle $\frac{z}{w}$ such that $\frac{z}{w} \neq \frac{1}{0}$ and $\frac{z}{w} \neq (\frac{c_i}{d_i} + e_2)$ for any i where e_2 is an arbitrary integral tangle. Then $N(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n} + \frac{x}{y} + \frac{z}{w}) = N(\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m} + \frac{x}{y} + \frac{z}{w})$. Then by theorem 5.4, $\frac{a_i}{b_i} = \frac{c_i}{d_i}$ for all i.

($\Leftarrow$) It is trivial. $\square$

Proposition 5.13. *Suppose that $0 < a_i < b_i$ for $1 \leq i \leq n$, h_j's have the same sign for all j, $h_j \neq 0$ for $2 \leq j \leq t-1$ and t is odd. For $n \geq 2$, $(\frac{-b_1}{a_1} * \cdots * \frac{-b_n}{a_n}) \circ (-h_1, \cdots, -h_t, 0) = (\frac{-d_1}{c_1} * \cdots * \frac{-d_m}{c_m}) \circ (-k_1, \cdots, -k_s, 0)$ where $0 < c_i < d_i$, k_j's have the same sign for all j, $k_j \neq 0$ for $2 \leq j \leq s-1$ and s is odd iff (a) $n = m$ and $\frac{a_i}{b_i} = \frac{c_i}{d_i}$ for all i and (b) $t = s$ and $h_j = k_j$ for all j.*

Proof. ($\Rightarrow$) Suppose $(\frac{-b_1}{a_1} * \cdots * \frac{-b_n}{a_n}) \circ (-h_1, \cdots, -h_t, 0) = (\frac{-d_1}{c_1} * \cdots * \frac{-d_m}{c_m}) \circ (-k_1, \cdots, -k_s, 0)$. By rotating both these tangles about the lines connecting NW and SE endpoints of $(\frac{-b_1}{a_1} * \cdots * \frac{-b_n}{a_n}) \circ (-h_1, \cdots, -h_t, 0)$ and $(\frac{-d_1}{c_1} * \cdots * \frac{-d_m}{c_m}) \circ (-k_1, \cdots, -k_s, 0)$ respectively, we have $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t) = (\frac{c_1}{d_1} + \cdots + \frac{c_m}{d_m}) \circ (k_1, \cdots, k_s)$. By proposition 5.12, we have the result.

($\Leftarrow$) It is clear. $\square$

Proposition 5.14. *Suppose that $0 < a_i < b_i$ for $1 \leq i \leq n$, h_j's have the same sign for all j, $h_j \neq 0$ for $2 \leq j \leq t-1$ and t is odd. If $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t) = (\frac{-d_1}{c_1} * \cdots * \frac{-d_m}{c_m}) \circ (-k_1, \cdots, -k_s, 0)$ where $0 < c_i < d_i$, k_j's have the same sign for all j, $k_j \neq 0$ for $2 \leq j \leq s-1$ and s is odd, then $n = m = 1$. That is, $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t)$ is a rational tangle.*

Proof. Suppose $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t) = (\frac{-d_1}{c_1} * \cdots * \frac{-d_m}{c_m}) \circ (-k_1, \cdots, -k_s, 0)$. Then $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t, k_s, \cdots, k_1, 0) = (\frac{-d_1}{c_1} * \cdots * \frac{-d_m}{c_m})$. Taking numerator closure of both sides, $N((\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t, k_s, \cdots, k_1, 0)) = N(\frac{-d_1}{c_1} * \cdots * \frac{-d_m}{c_m}) = D(\frac{c_1}{d_1}) \sharp \cdots \sharp D(\frac{c_m}{d_m})$. Since $N((\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t, k_s, \cdots, k_1, 0)) = N((\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) + (0, k_1, \cdots, k_s, h_t, \cdots, h_1))$ is either a rational link or a Montesinos link, $m = 1$. Then $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t) = \frac{-d_1}{c_1} \circ (-k_1, \cdots, -k_s, 0)$ which is a rational tangle. Thus $n = 1$. That is, $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t) = \frac{a_1}{b_1} \circ (h_1, \cdots, h_t)$. $\square$

Theorems 5.2, 5.3, propositions 5.12, 5.13 and 5.14 give the following classification of generalized Montesinos tangles which are not rational tangles.

Theorem 5.15. (Classification of Generalized Montesinos Tangles)

A generalized Montesinos tangle which is not rational is uniquely represented as one of the following:

(1) $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t)$ *when* $n \geq 2$,

(2) $(\frac{-b_1}{a_1} * \cdots * \frac{-b_n}{a_n}) \circ (-h_1, \cdots, -h_t, 0)$ *when* $n \geq 2$.

where $0 < a_i < b_i$, t *is odd,* h_j*'s have the same sign for all* j *and* $h_j \neq 0$ *for* $2 \leq j \leq t-1$

Proof. By theorems 5.2 and 5.3, a generalized Montesinos tangle has the form $(\frac{a_1}{b_1} + \cdots + \frac{a_n}{b_n}) \circ (h_1, \cdots, h_t)$ or $(\frac{-b_1}{a_1} * \cdots * \frac{-b_n}{a_n}) \circ (-h_1, \cdots, -h_t, 0)$ where $0 < a_i < b_i$, t is odd, h_j's have the same sign for all j and $h_j \neq 0$ for $2 \leq j \leq t-1$. By proposition 5.12 and 5.13, this is unique. $\square$

The sum of two rational tangles need not be rational but the numerator closure of the sum of two rational tangles is a rational knot.

Lemma 5.16. *[20]* $N(\frac{j}{p} + \frac{t}{w}) = N(\frac{jw + pt}{dw + qt})$ *where* d *and* q *are any integers such that* $pd - qj = 1$.

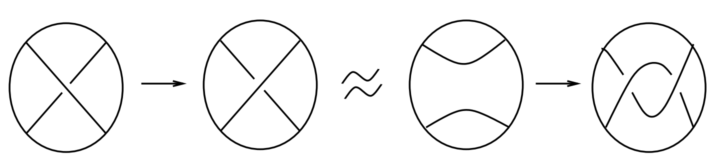
Lemma 5.17. *[12] If* $N(\frac{j}{p}+\frac{f}{g}) = N(\frac{a}{b})$*, then* $\frac{f}{g} = \frac{da - jb'}{pb' - qa}$ *for some integers* d*,* q*, and* b' *such that* $pd - qj = 1$*,* $b'b^{\pm 1} = 1$ *mod* a*.*

5.2 Equivalent moves

Definition 5.6. [12] If there is a solution for U such that $N(U + B) = K_1$ and $N(U + E) = K_2$, then K_2 is said to be obtained from K_1 by a (B, E)-move.

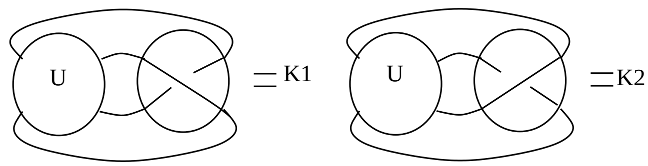
Definition 5.7. [12] A (B, E)-move is said to be equivalent to a (B', E')-move if there exists a solution for U such that $N(U + B) = K_1$ and $N(U + E) = K_2$ if and only if there exists a solution for U' such that $N(U' + B') = K_1$ and $N(U' + E') = K_2$. The above two systems of tangle equations are said to be equivalent.

Figure 5.7 and Figure 5.8 shows that $(+1, -1)$-move is equivalent to $(0, -2)$-move. Generally, $(\frac{f_1}{g_1}, \frac{f_2}{g_2})$-move is equivalent to $(0, \frac{x}{y})$-move where $\frac{f_1}{g_1} = (c_1, \cdots, c_n)$ and $\frac{x}{y} = (\frac{f_2}{g_2}) \circ (-c_n, \cdots, -c_1)$ [12]. Thus it is sufficient to solve $N(U + \frac{0}{1}) = K_1$ and $N(U + \frac{x}{y}) = K_2$ instead of solving $N(U + \frac{f_1}{g_1}) = K_1$ and $N(U + \frac{f_2}{g_2}) = K_2$.

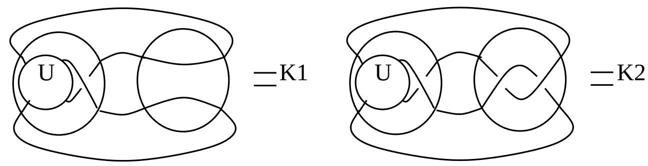


(+1) (−1) (0) (−2)

Figure 5.7: Equivalent moves.



(a) $N(U + (1)) = K1$ $\qquad$ $N(U + (-1)) = K2$



(b) $N(U \circ (1) + (0)) = K1$ $\qquad$ $N(U \circ (1) + (-2)) = K2$

Figure 5.8: Equivalent tangle equations.

5.3 Solving tangle equations

First, the system of tangle equations is solved when $s \geq 3$ and $t \geq 3$. That is, the righthand side of equations (1) and (2) are both Montesinos links.

Theorem 5.18. *Suppose that a_i, b_i, e_1, x, y are integers and $0 < a_i < b_i$ for $1 \leq i \leq s$. For $s, t \geq 3$,*

$$N(U + \frac{0}{1}) = N(\frac{a_1}{b_1} + \cdots + \frac{a_s}{b_s} + e_1) \quad (1)$$

$$\text{and } N(U + \frac{x}{y}) = N(\frac{z_1}{v_1} + \cdots + \frac{z_t}{v_t} + e_2) \quad (2)$$

where z_j, v_j, e_2 are integers and $0 < z_j < v_j$ for $1 \leq j \leq t$

and $U = (\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) \circ (h_1, \cdots, h_m)$ is a generalized M-tangle.

if and only if for $s, t \geq 3$ and for some $1 \leq j_1 \leq s$,

1. *If* $m=1$, *then* $U=(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{a_{j_1\mp1}}{b_{j_1\mp1}})\circ(e_1)$ *and* $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2)=N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{a_{j_1\mp1}}{b_{j_1\mp1}}+\frac{x}{y}+e_1)$ *where* $\frac{x}{y}=e_2-e_1$ *and* $t=s$ *if* $\frac{x}{y}$ *is an integer and* $\frac{x}{y}=\frac{z_k}{v_k}+e_2-e_1$ *for some* k *and* $t=s+1$ *if* $\frac{x}{y}$ *is not an integer.*

2. *If* $m=3$ *and* $h_m=0$, *then* $U=(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{a_{j_1\mp1}}{b_{j_1\mp1}})\circ(e_1,h_2,0)$ *and* $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2)=N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{a_{j_1\mp1}}{b_{j_1\mp1}}+\frac{x}{h_2x+y}+e_1)$ *where* $h_2=\frac{1}{e_2-e_1}-\frac{y}{x}$ *and* $t=s$ *if* $\frac{x}{h_2x+y}$ *is an integer and* $h_2=\frac{v_k}{z_k+(e_2-e_1)v_k}-\frac{y}{x}$ *for some* k *and* $t=s+1$ *if* $\frac{x}{h_2x+y}$ *is not an integer.*

3. *If* $m>3$ *or if* $m=3$ *and* $h_m\neq0$, *then assuming that* h_i*'s have the same sign and* $h_i\neq0$ *for* $2\leq i\leq m-1$,

   (a) $U=(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp3}}{b_{j_1\mp3}}+\frac{a_{j_1\mp2}}{b_{j_1\mp2}})\circ(h_1,\cdots,h_m)$ *and* $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2)=N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp3}}{b_{j_1\mp3}}+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{xE[h_1,\cdots,h_{m-1}]+yE[h_1,\cdots,h_m]}{xE[h_2,\cdots,h_{m-1}]+yE[h_2,\cdots,h_m]})$ *where* $\frac{E[h_1,\cdots,h_m]}{E[h_2,\cdots,h_m]}=\frac{a_{j_1\mp1}}{b_{j_1\mp1}}+e_1$ *for some* j_1.

   (b) $U=(\frac{-b_{j_1}}{a_{j_1}}*\frac{-b_{j_1\pm1}}{a_{j_1\pm1}}*\cdots*\frac{-b_{j_1\mp3}}{a_{j_1\mp3}}*\frac{-b_{j_1\mp2}}{a_{j_1\mp2}})\circ(-h_1,\cdots,-h_m,0)$ *and* $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2)=N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp3}}{b_{j_1\mp3}}+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{yE[h_1,\cdots,h_{m-1}]-xE[h_1,\cdots,h_m]}{yE[h_2,\cdots,h_{m-1}]-xE[h_2,\cdots,h_m]})$ *where* $h_2\neq\pm1$ *and* $\frac{E[h_1,\cdots,h_{m-1}]}{E[h_2,\cdots,h_{m-1}]}=\frac{a_{j_1\mp1}}{b_{j_1\mp1}}+e_1$ *for some* j_1.

   (c) $m=3$ *and* $U=(\frac{-b_{j_1}}{a_{j_1}}*\frac{-b_{j_1\pm1}}{a_{j_1\pm1}}*\cdots*\frac{-b_{j_1\mp2}}{a_{j_1\mp2}}*\frac{-b_{j_1\mp1}}{a_{j_1\mp1}})\circ(-h_1,\mp1,-h_3,0)$ *and* $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2)=N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{a_{j_1\mp1}}{b_{j_1\mp1}}+\frac{y(1\pm h_1)-x(h_1+h_3\pm h_1h_3)}{\pm y-x(1\pm h_3)})$ *where* $h_1=e_1\mp1$.

Note that if $h_2 = 0$ in Case 2, the solution is the same as the solution in Case 1. We can consider only when m is odd since $T \circ (h_1, \cdots, h_{2k}) = T \circ (0, h_1, \cdots, h_{2k})$ for a tangle T.

Proof. If U is rational, $N(U)$ is a 4-plat but $N(\frac{a_1}{b_1} + \cdots + \frac{a_s}{b_s} + e_1)$ is a Montesinos link/knot by equation (1). It contradicts. So U is not rational. Since U is a generalized M-tangle which is not rational, it can be written as $U = (\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) \circ (h_1, \cdots, h_m)$ where c_i, d_i and h_j are integers such that $0 < c_i < d_i$ for $1 \le i \le n$, $1 \le j \le m$ and $n \ge 2$. W.L.O.G, we can assume m is odd.

Case 1: If $m = 1$, from equation (1),

$$\begin{aligned} N(U + \frac{0}{1}) =& N((\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) \circ (h_1) + (0)) \\ =& N(\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n} + h_1) \\ =& N(\frac{a_1}{b_1} + \cdots + \frac{a_s}{b_s} + e_1). \end{aligned}$$

Then by theorem 5.4, $h_1 = e_1$, $\frac{c_i}{d_i} = \frac{a_j}{b_j}$ with $n = s$ where $i = 1, \cdots, n$, $j = j_1, j_1 + 1, \cdots, s, 1, 2, \cdots, j_1 - 2, j_1 - 1$ or $j = j_1, j_1 - 1, \cdots, 2, 1, s, \cdots, j_1 + 2, j_1 + 1$ for some $1 \le j_1 \le s$. Here, $n \ge 3$ since $s \ge 3$.

From equation (2),

$$\begin{aligned} N(U + \frac{x}{y}) =& N((\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) \circ (h_1) + \frac{x}{y}) \\ =& N((\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) + \frac{x}{y} \circ (h_1)) \quad \text{by lemma 5.6} \\ =& N(\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n} + \frac{x}{y} + h_1) \\ =& N(\frac{z_1}{v_1} + \cdots + \frac{z_t}{v_t} + e_2). \end{aligned}$$

Then by the above result, $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2) = N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm 1}}{b_{j_1\pm 1}}+\cdots+\frac{a_{j_1\mp 2}}{b_{j_1\mp 2}}+\frac{a_{j_1\mp 1}}{b_{j_1\mp 1}}+\frac{x}{y}+e_1)$. By theorem 5.4, $e_2 = \frac{x}{y}+e_1$ and $t = s$ if $\frac{x}{y}$ is an integer and $\frac{z_k}{v_k}+e_2 = \frac{x}{y}+e_1$ for some k and $t = s+1$ if $\frac{x}{y}$ is not an integer.

Case 2: Assume that $m = 3$ and $h_m = 0$. Then by equation(1),

$$\begin{aligned}
N(U+\frac{0}{1}) =& N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})\circ(h_1,h_2,0)+(0)) \\
=& N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})\circ(h_1)) \\
=& N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+h_1) \\
=& N(\frac{a_1}{b_1}+\cdots+\frac{a_s}{b_s}+e_1).
\end{aligned}$$

Then by theorem 5.4, $h_1 = e_1$, $\frac{c_i}{d_i} = \frac{a_j}{b_j}$ with $n = s$ where $i = 1,\cdots,n$, $j = j_1, j_1+1,\cdots,s,1,2,\cdots,j_1-2,j_1-1$ or $j = j_1,j_1-1,\cdots,2,1,s,\cdots,j_1+2,j_1+1$ for some $1 \le j_1 \le s$. Here, $n \ge 3$.

From equation (2),

$$\begin{aligned}
N(U+\frac{x}{y}) =& N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})\circ(h_1,h_2,0)+\frac{x}{y}) \\
=& N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})+(\frac{x}{y})\circ(0,h_2,h_1)) \quad \text{by lemma 5.6} \\
=& N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})+(\frac{x}{y})\circ(0,h_2,0)+h_1) \\
=& N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})+\frac{1}{h_2+\frac{1}{\frac{x}{y}}}+h_1) \\
=& N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+\frac{x}{h_2x+y}+h_1) \\
=& N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2).
\end{aligned}$$

Then by the above result, $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2) = N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm 1}}{b_{j_1\pm 1}}+\cdots+\frac{a_{j_1\mp 2}}{b_{j_1\mp 2}}+$

$\frac{a_{j_1\mp1}}{b_{j_1\mp1}} + \frac{x}{h_2x+y} + e_1)$. By theorem 5.4, $\frac{x}{h_2x+y} + e_1 = e_2$ and $t = s$ if $\frac{x}{h_2x+y}$ is an integer and $\frac{x}{h_2x+y} + e_1 = \frac{z_k}{v_k} + e_2$ for some k and $t = s+1$ if $\frac{x}{h_2x+y}$ is not an integer.

Case 3: Assume that $m > 3$ or $m = 3$ and $h_m \neq 0$. If we assume that h_j's have the same sign and $h_j \neq 0$ for $2 \leq j \leq m-1$, then by theorem 5.15, $U = (\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) \circ (h_1, \cdots, h_m)$ or $(\frac{-d_1}{c_1} * \cdots * \frac{-d_n}{c_n}) \circ (-h_1, \cdots, -h_m, 0)$. Since $N(U)$ is a Montesinos link by equation (1), case (3) in theorem 5.15 is ruled out.

(1) If $U = (\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) \circ (h_1, \cdots, h_m)$ where $0 < c_i < d_i$, m is odd, h_j's have the same sign, $h_j \neq 0$ for $2 \leq j \leq m-1$, then by equation (1),

$$
\begin{aligned}
N(U + \frac{0}{1}) =& N((\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) \circ (h_1, \cdots, h_m) + (0)) \\
=& N((\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) + (0) \circ (h_m, \cdots, h_1)) \quad \text{by lemma 5.6} \\
=& N((\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) + (h_m, \cdots, h_1)) \\
=& N((\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n}) + h_1 + \frac{1}{h_2 + \cdots + \frac{1}{h_m}}) \\
=& N(\frac{c_1}{d_1} + \cdots + \frac{c_n}{d_n} + \frac{E[h_1, \cdots, h_m]}{E[h_2, \cdots, h_m]}) \quad \text{by proposition 5.9} \\
=& N(\frac{a_1}{b_1} + \cdots + \frac{a_s}{b_s} + e_1).
\end{aligned}
$$

Since h_j's have the same sign, m is odd, $m \geq 3$ and $h_m \neq 0$ if $m = 3$, $\frac{E[h_1, \cdots, h_m]}{E[h_2, \cdots, h_m]}$ cannot be an integer. Then by theorem 5.4, $\frac{c_i}{d_i} = \frac{a_j}{b_j}$ with $n = s - 1$ where $i = 1, \cdots, n$, $j = j_1, j_1 + 1, \cdots, s, 1, 2, \cdots, j_1 - 2$ or $j = j_1, j_1 - 1, \cdots, 2, 1, s, \cdots, j_1 + 2$ and $\frac{E[h_1, \cdots, h_m]}{E[h_2, \cdots, h_m]} = \frac{a_{j_1\mp1}}{b_{j_1\mp1}} + e_1$ for some $1 \leq j_1 \leq s$. Here, $n \geq 2$.

From equation (2),

$$\begin{aligned}
N(U+\frac{x}{y}) =& N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})\circ(h_1,\cdots,h_m)+(l_1,\cdots,l_k)) \\
& \text{where } \frac{x}{y}=(l_1,\cdots,l_k). \\
=& N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})+(l_1,\cdots,l_k)\circ(h_m,\cdots,h_1)) \quad \text{by lemma 5.6} \\
=& N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+(l_1,\cdots,l_k+h_m,\cdots,h_1)) \quad \text{by lemma 5.8} \\
=& N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+[h_1,h_2,\cdots,h_m+l_k,\cdots,l_2,l_1]) \\
=& N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+\frac{xE[h_1,\cdots,h_{m-1}]+yE[h_1,\cdots,h_m]}{xE[h_2,\cdots,h_{m-1}]+yE[h_2,\cdots,h_m]}) \\
& \text{by proposition 5.9 and lemma 5.10} \\
=& N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2).
\end{aligned}$$

Then by the above result, $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2) = N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp3}}{b_{j_1\mp3}}+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{xE[h_1,\cdots,h_{m-1}]+yE[h_1,\cdots,h_m]}{xE[h_2,\cdots,h_{m-1}]+yE[h_2,\cdots,h_m]})$ where $\frac{E[h_1,\cdots,h_m]}{E[h_2,\cdots,h_m]} = \frac{a_{j_1\mp1}}{b_{j_1\mp1}}+e_1$ for some $1\le j_1\le s$.

(2) If $U=(\frac{-d_1}{c_1}*\cdots*\frac{-d_n}{c_n})\circ(-h_1,\cdots,-h_m,0)$ where $0<c_i<d_i$, m is odd, h_j's have the same sign, $h_j\neq0$ for $2\le j\le m-1$, then by equation (1),

$$\begin{aligned}
N(U+\frac{0}{1}) =& N((\frac{-d_1}{c_1}*\cdots*\frac{-d_n}{c_n})\circ(-h_1,\cdots,-h_m,0)) \\
=& D((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})\circ(h_1,\cdots,h_m)) \\
=& N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})\circ(h_1,\cdots,h_m)+(0,0)) \\
=& N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+(0,0)\circ(h_m,\cdots,h_1)) \quad \text{by lemma 5.6}
\end{aligned}$$

$$
\begin{aligned}
&=N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+(h_{m-1},\cdots,h_1))\\
&=N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+h_1+\frac{1}{h_2+\cdots+\frac{1}{h_{m-1}}})\\
&=N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+\frac{E[h_1,\cdots,h_{m-1}]}{E[h_2,\cdots,h_{m-1}]})\ \text{ by proposition 5.9}\\
&=N(\frac{a_1}{b_1}+\cdots+\frac{a_s}{b_s}+e_1).
\end{aligned}
$$

If $m>3$ or $m=3$ and $h_2\neq\pm1$, $\dfrac{E[h_1,\cdots,h_{m-1}]}{E[h_2,\cdots,h_{m-1}]}$ cannot be an integer since h_j's have the same sign. Then by theorem 5.4, $\dfrac{c_i}{d_i}=\dfrac{a_j}{b_j}$ with $n=s-1$ where $i=1,\cdots,n$, $j=j_1,j_1{+}1,\cdots,s,1,2,\cdots,j_1{-}2$ or $j=j_1,j_1{-}1,\cdots,2,1,s,\cdots,j_1{+}2$ and $\dfrac{E[h_1,\cdots,h_{m-1}]}{E[h_2,\cdots,h_{m-1}]}=\dfrac{a_{j_1\mp1}}{b_{j_1\mp1}}+e_1$ for some $1\le j_1\le s$. Here, $n\ge2$. If $m=3$ and $h_2=\pm1$, then $\dfrac{E[h_1,h_2]}{E[h_2]}=\dfrac{h_1h_2+1}{h_2}=h_1\pm1$. By theorem 5.4, $\dfrac{c_i}{d_i}=\dfrac{a_j}{b_j}$ with $n=s$ where $i=1,\cdots,n$, $j=j_1,j_1{+}1,\cdots,s,1,2,\cdots,j_1{-}1$ or $j=j_1,j_1{-}1,\cdots,2,1,s,\cdots,j_1{+}1$ and $h_1\pm1=e_1$.

From equation (2),

$$N(U+\frac{x}{y})=N((\frac{-d_1}{c_1}*\cdots*\frac{-d_n}{c_n})\circ(-h_1,\cdots,-h_m,0)+(l_1,\cdots,l_k))$$

where $\dfrac{x}{y}=(l_1,\cdots,l_k)$ and k is odd,

$$
\begin{aligned}
&=N((\frac{-d_1}{c_1}*\cdots*\frac{-d_n}{c_n})\circ(-h_1,\cdots,-h_m,0)\circ(l_k,\cdots,l_1))\ \text{ by lemma 5.6}\\
&=N((\frac{-d_1}{c_1}*\cdots*\frac{-d_n}{c_n})\circ(-h_1,\cdots,-h_m,l_k,\cdots,l_1))\ \text{ by lemma 5.8}\\
&=D((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})\circ(h_1,\cdots,h_m,-l_k,\cdots,-l_1,0))
\end{aligned}
$$

$$=N((\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n})\circ(h_1,\cdots,h_m,-l_k,\cdots,-l_1,0)+(0,0))$$

$$=N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+(0,0)\circ(0,-l_1,\cdots,-l_k,h_m,\cdots,h_1)) \quad \text{by lemma 5.6}$$

$$=N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+(-l_1,\cdots,-l_k,h_m,\cdots,h_1))$$

$$=N(\frac{c_1}{d_1}+\cdot+\frac{c_n}{d_n}+\frac{E[h_1,\cdots,h_m,0]E[-l_1,\cdots,-l_{k-1}]+E[h_1,\cdots,h_m]E[-l_1,\cdots,-l_k]}{E[h_2,\cdots,h_m,0]E[-l_1,\cdots,-l_{k-1}]+E[h_2,\cdots,h_m]E[-l_1,\cdots,-l_k]})$$

$$=N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+\frac{yE[h_1,\cdots,h_m,0]-xE[h_1,\cdots,h_m]}{yE[h_2,\cdots,h_m,0]-xE[h_2,\cdots,h_m]})$$

$$=N(\frac{c_1}{d_1}+\cdots+\frac{c_n}{d_n}+\frac{yE[h_1,\cdots,h_{m-1}]-xE[h_1,\cdots,h_m]}{yE[h_2,\cdots,h_{m-1}]-xE[h_2,\cdots,h_m]})$$

by proposition 5.9 and lemma 5.10

$$=N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2).$$

By the above result, if $m>3$ or $m=3$ and $h_2\neq\pm1$, then $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2)=N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp3}}{b_{j_1\mp3}}+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{yE[h_1,\cdots,h_{m-1}]-xE[h_1,\cdots,h_m]}{yE[h_2,\cdots,h_{m-1}]-xE[h_2,\cdots,h_m]})$ where $\frac{E[h_1,\cdots,h_{m-1}]}{E[h_2,\cdots,h_{m-1}]}=\frac{a_{j_1\mp1}}{b_{j_1\mp1}}+e_1$ for some $1\leq j_1\leq s$. If $m=3$ and $h_2=\pm1$, then $N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2)=N(\frac{a_{j_1}}{b_{j_1}}+\frac{a_{j_1\pm1}}{b_{j_1\pm1}}+\cdots+\frac{a_{j_1\mp2}}{b_{j_1\mp2}}+\frac{a_{j_1\mp1}}{b_{j_1\mp1}}+\frac{y(1\pm h_1)-x(h_1+h_3\pm h_1h_3)}{\pm y-x(1\pm h_3)})$ where $h_1=e_1\mp1$. $\square$

Next, the system of tangle equations is solved when $s\leq2$ and $t\geq3$. If $s\leq2$, then the righthand side of equation (1) in theorem 5.18 is a rational link. So it can be written as $N(\frac{a}{b})$.

Theorem 5.19. *Suppose that a, b, x, y are integers. For $t \geq 3$,*

$$N(U + \frac{0}{1}) = N(\frac{a}{b}) \quad (1)$$

$$\text{and } N(U + \frac{x}{y}) = N(\frac{z_1}{v_1} + \cdots + \frac{z_t}{v_t} + e_2) \quad (2)$$

where z_j, v_j, e_2 are integers and $0 < z_j < v_j$ for $1 \leq j \leq t$

and U is a generalized M-tangle.

if and only if $t = 3$,

$U = (\frac{c_1}{d_1} + \frac{pa - c_1 b}{d_1 b - qa}) \circ (h, 0)$ and $(\frac{pa - c_1 b}{d_1 b - qa} + \frac{c_1}{d_1}) \circ (h, 0)$ for all integers c_1, d_1, p and q such that $d_1 p - qc_1 = 1$ and $0 < c_1 < d_1$ where $(\frac{c_1}{d_1}, \frac{pa - c_1 b}{d_1 b - qa}, \frac{x}{hx + y}) = (\frac{z_{i_1}}{v_{i_1}}, \frac{z_{i_2}}{v_{i_2}} + k, \frac{z_{i_3}}{v_{i_3}} + e_2 - k)$ for some integer k where $\{i_1, i_2, i_3\}$ is cyclic permutations of $(1, 2, 3)$ and reversal of order.

Note that the choice of c_1 and p such that $d_1 p - qc_1 = 1$ has no effect on U.

Proof. If U is rational, $N(U + \frac{x}{y})$ is a rational link but $N(\frac{z_1}{v_1} + \cdots + \frac{z_t}{v_t} + e_2)$ is a Montesinos link/knot by the equation (2). It contradicts. So U should be a generalized M-tangle which is not rational. Since $N(U) = N(\frac{a}{b})$ is rational by equation (1), $U = (\frac{c_1}{d_1} + \frac{c_2}{d_2}) \circ (h, 0)$ where $0 < c_i < d_i$ for $i = 1, 2$. From the proof of theorem 3 in [12], $U = (\frac{c_1}{d_1} + \frac{pa - c_1 b}{d_1 b - qa}) \circ (h, 0)$ and $(\frac{pa - c_1 b}{d_1 b - qa} + \frac{c_1}{d_1}) \circ (h, 0)$ for integers c_1, d_1, p and q such that $d_1 p - qc_1 = 1$. Note that if d_1 and q are specified, then the choice of c_1 and p such that $d_1 p - qc_1 = 1$ has no effect on U since $\frac{c_1}{d_1} + \frac{pa - c_1 b}{d_1 b - qa} = \frac{c_1 + d_1 i}{d_1} + \frac{pa - c_1 b - (d_1 b - qa)i}{d_1 b - qa} = \frac{c_1 + d_1 i}{d_1} + \frac{(p + qi)a - (c_1 + d_1 i)b}{d_1 b - qa}$ and $d_1(p + qi) - q(c_1 + d_1 i) = 1$ if and only if $d_1 p - qc_1 = 1$ [12].

From equation (2),

$$\begin{aligned}N(U+\frac{x}{y}) =& N((\frac{c_1}{d_1}+\frac{c_2}{d_2})\circ(h,0)+\frac{x}{y})\\ =& N((\frac{c_1}{d_1}+\frac{c_2}{d_2})+(\frac{x}{y})\circ(h,0))\\ =& N(\frac{c_1}{d_1}+\frac{c_2}{d_2}+\frac{x}{hx+y})\\ =& N(\frac{z_1}{v_1}+\cdots+\frac{z_t}{v_t}+e_2).\end{aligned}$$

Since $t \geq 3$, $\frac{x}{hx+y}$ cannot be an integer. Actually, $t = 3$ and $N(\frac{z_1}{v_1}+\frac{z_2}{v_2}+\frac{z_3}{v_3}+e_2) = N(\frac{c_1}{d_1}+\frac{pa-c_1b}{d_1b-qa}+\frac{x}{hx+y})$. Similarly, for $U = (\frac{pa-c_1b}{d_1b-qa}+\frac{c_1}{d_1})\circ(h,0)$, $N(\frac{z_1}{v_1}+\frac{z_2}{v_2}+\frac{z_3}{v_3}+e_2) = N(\frac{pa-c_1b}{d_1b-qa}+\frac{c_1}{d_1}+\frac{x}{hx+y})$ by equation (2). Note that by theorem 5.4, $N(\frac{pa-c_1b}{d_1b-qa}+\frac{c_1}{d_1}+\frac{x}{hx+y}) = N(\frac{c_1}{d_1}+\frac{pa-c_1b}{d_1b-qa}+\frac{x}{hx+y})$. Thus $(\frac{c_1}{d_1},\frac{pa-c_1b}{d_1b-qa},\frac{x}{hx+y}) = (\frac{z_{i_1}}{v_{i_1}},\frac{z_{i_2}}{v_{i_2}}+k,\frac{z_{i_3}}{v_{i_3}}+e_2-k)$ for some integer k where $\{i_1,i_2,i_3\}$ is cyclic permutations of $(1,2,3)$ and reversal of order by theorem 5.4. $\square$

Next, the system of tangle equations is solved when $s \geq 3$ and $t \leq 2$. If $t \leq 2$, then the righthand side of equation (2) in theorem 5.18 is a rational link. So it can be written as $N(\frac{z}{v})$.

Theorem 5.20. *Suppose that x, y, z, v are integers. For $s \geq 3$,*

$$N(U+\frac{0}{1}) = N(\frac{a_1}{b_1}+\cdots+\frac{a_s}{b_s}+e_1) \quad (1)$$

$$\textit{and } N(U+\frac{x}{y}) = N(\frac{z}{v}) \quad (2)$$

where a_i, b_i, e_1 are integers and $0 < a_i < b_i$ for $1 \leq i \leq s$,

and U is a generalized M-tangle.

if and only if $s = 3$,

$U = (\frac{c_1}{d_1} + \frac{pz - c_1 v}{d_1 v - qz}) \circ (h, 0) \circ (-\frac{x}{y})$ *and* $(\frac{pz - c_1 v}{d_1 v - qz} + \frac{c_1}{d_1}) \circ (h, 0) \circ (-\frac{x}{y})$ *for integers* c_1, d_1, p *and* q *such that* $d_1 p - qc_1 = 1$ *and* $0 < c_1 < d_1$ *where* $(\frac{c_1}{d_1}, \frac{pz - c_1 v}{d_1 v - qz}, \frac{x}{hx - y'}) = (\frac{a_{i_1}}{b_{i_1}}, \frac{a_{i_2}}{b_{i_2}} + k, \frac{a_{i_3}}{b_{i_3}} + e_1 - k)$ *for some integer* k *where* $\{i_1, i_2, i_3\}$ *is cyclic permutations of* $(1, 2, 3)$ *and reversal of order and* y' *such that* $yy'^{\pm 1} = 1$ *mod* x.

Note that the choice of c_1 *and* p *such that* $d_1 p - qc_1 = 1$ *has no effect on* U.

Proof. If U is rational, then $N(U)$ is a rational link but by equation (1), $N(\frac{a_1}{b_1} + \cdots + \frac{a_s}{b_s} + e_1)$ is a Montesinos knot. It contradicts. So U is a nonrational generalized M-tangle.

By equation (2),

$$
\begin{aligned}
N(U + \frac{x}{y}) =& N(U + (l_1, \cdots, l_k)) \quad \text{where } \frac{x}{y} = (l_1, \cdots, l_k) \text{ and } k \text{ is odd} \\
=& N(U \circ (l_k, \cdots, l_1) + (\frac{0}{1})) \quad \text{by lemma 5.6} \\
=& N(\frac{z}{v})
\end{aligned}
$$

Since U is a nonrational generalized M-tangle, so is $U \circ (l_k, \cdots, l_1)$. Since $N(U \circ (l_k, \cdots, l_1)) = N(\frac{z}{v})$ is rational, we can write $U \circ (l_k, \cdots, l_1) = (\frac{c_1}{d_1} + \frac{c_2}{d_2}) \circ (h, 0)$ where $0 < c_i < d_i$ for $i = 1, 2$. From the proof of theorem 3 in [12], $U \circ (l_k, \cdots, l_1) = (\frac{c_1}{d_1} + \frac{pz - c_1 v}{d_1 v - qz}) \circ (h, 0)$ and $(\frac{pz - c_1 v}{d_1 v - qz} + \frac{c_1}{d_1}) \circ (h, 0)$ for integers c_1, d_1, p and q such that $d_1 p - qc_1 = 1$. Thus $U = (\frac{c_1}{d_1} + \frac{pz - c_1 v}{d_1 v - qz}) \circ (h, 0) \circ (-l_1, \cdots, -l_k) = (\frac{c_1}{d_1} + \frac{pz - c_1 v}{d_1 v - qz}) \circ (h, 0) \circ (-\frac{x}{y})$ and $U = (\frac{pz - c_1 v}{d_1 v - qz} + \frac{c_1}{d_1}) \circ (h, 0) \circ (-\frac{x}{y})$ for integers c_1, d_1, p and q satisfying the above. Note that if d_1 and q are specified, then the choice of c_1 and p such that $d_1 p - qc_1 = 1$ has no effect on U.

By equation(1),

$$
\begin{aligned}
N(U+\frac{0}{1}) =& N((\frac{c_1}{d_1}+\frac{pz-c_1v}{d_1v-qz})\circ(h,0)\circ(-l_1,\cdots,-l_k)+(0)) \\
=& N((\frac{c_1}{d_1}+\frac{pz-c_1v}{d_1v-qz})\circ(h,0)+(-l_k,\cdots,-l_1)) \text{ by lemma 5.6} \\
=& N((\frac{c_1}{d_1}+\frac{pz-c_1v}{d_1v-qz})\circ(h,0)+(-\frac{x}{y'})) \quad \text{where } yy'^{\pm 1}=1 \bmod x \\
=& N((\frac{c_1}{d_1}+\frac{pz-c_1v}{d_1v-qz})+(-\frac{x}{y'})\circ(h,0)) \\
=& N(\frac{c_1}{d_1}+\frac{pz-c_1v}{d_1v-qz}+\frac{x}{hx-y'}) \\
=& N(\frac{a_1}{b_1}+\cdots+\frac{a_s}{b_s}+e_1)
\end{aligned}
$$

Since $s \geq 3$, $\frac{x}{hx-y'}$ cannot be an integer. Actually, $s=3$ and $N(\frac{a_1}{b_1}+\frac{a_2}{b_2}+\frac{a_3}{b_3}+e_1) = N(\frac{c_1}{d_1}+\frac{pz-c_1v}{d_1v-qz}+\frac{x}{hx-y'})$. Similarly, for $U=(\frac{pz-c_1v}{d_1v-qz}+\frac{c_1}{d_1})\circ(h,0)$, $N(\frac{a_1}{b_1}+\frac{a_2}{b_2}+\frac{a_3}{b_3}+e_1) = N(\frac{pz-c_1v}{d_1v-qz}+\frac{c_1}{d_1}+\frac{x}{hx-y'})$. Note that by theorem 5.4, $N(\frac{pz-c_1v}{d_1v-qz}+\frac{c_1}{d_1}+\frac{x}{hx-y'}) = N(\frac{c_1}{d_1}+\frac{pz-c_1v}{d_1v-qz}+\frac{x}{hx-y'})$. Thus $(\frac{c_1}{d_1}, \frac{pz-c_1v}{d_1v-qz}, \frac{x}{hx-y'}) = (\frac{a_{i_1}}{b_{i_1}}, \frac{a_{i_2}}{b_{i_2}}+k, \frac{a_{i_3}}{b_{i_3}}+e_1-k)$ for some integer k where $\{i_1, i_2, i_3\}$ is cyclic permutations of $(1,2,3)$ and reversal of order by theorem 5.4. $\square$

Example 5.1. Solve the following tangle equations where U is a generalized M-tangle.

$$
\begin{aligned}
& N(U+\frac{0}{1}) = N(\frac{1}{2}+\frac{2}{3}+\frac{2}{3}+(-3)) \\
\text{and } \ & N(U+\frac{x}{y}) = N(\frac{1}{2}+\frac{3}{5}+\frac{2}{3}+(-3)).
\end{aligned}
$$

By theorem 5.18, if $m = 1$, then $U = (\frac{1}{2} + \frac{2}{3} + \frac{2}{3}) \circ (-3), (\frac{2}{3} + \frac{2}{3} + \frac{1}{2}) \circ (-3)$ or $(\frac{2}{3} + \frac{1}{2} + \frac{2}{3}) \circ (-3)$. Moreover, $N(\frac{1}{2} + \frac{3}{5} + \frac{2}{3} + (-3)) = N((\frac{1}{2} + \frac{2}{3} + \frac{2}{3}) + \frac{x}{y} + (-3))$, $N((\frac{2}{3} + \frac{2}{3} + \frac{1}{2}) + \frac{x}{y} + (-3))$ or $N((\frac{2}{3} + \frac{1}{2} + \frac{2}{3}) + \frac{x}{y} + (-3))$. By theorem 5.4, there is no $\frac{x}{y}$ satisfying the above.

Similarly, if $m = 3$ and $h_m = 0$, then by theorem 5.18, $U = (\frac{1}{2} + \frac{2}{3} + \frac{2}{3}) \circ (-3, h_2, 0), (\frac{2}{3} + \frac{2}{3} + \frac{1}{2}) \circ (-3, h_2, 0)$ or $(\frac{2}{3} + \frac{1}{2} + \frac{2}{3}) \circ (-3, h_2, 0)$. Moreover, $N(\frac{1}{2} + \frac{3}{5} + \frac{2}{3} + (-3)) = N((\frac{1}{2} + \frac{2}{3} + \frac{2}{3}) + \frac{x}{h_2 x + y} + (-3))$, $N((\frac{2}{3} + \frac{2}{3} + \frac{1}{2}) + \frac{x}{h_2 x + y} + (-3))$ or $N((\frac{2}{3} + \frac{1}{2} + \frac{2}{3}) + \frac{x}{h_2 x + y} + (-3))$ which are impossible by theorem 5.4.

Thus $m > 3$ or $m = 3$ and $h_m \neq 0$.

(1) By theorem 5.18 (3a), $U = (\frac{1}{2} + \frac{2}{3}) \circ (h_1, \cdots, h_m)$ or $(\frac{2}{3} + \frac{1}{2}) \circ (h_1, \cdots, h_m)$where $\frac{E[h_1, \cdots, h_m]}{E[h_2, \cdots, h_m]} = \frac{2}{3} + (-3) = \frac{-7}{3} = -2 + \frac{1}{-2 + \frac{1}{-1}}$. This implies that $m = 3$ and $(h_1, h_2, h_3) = (-2, -2, -1)$. Moreover, $N(\frac{1}{2} + \frac{3}{5} + \frac{2}{3} + (-3)) = N(\frac{1}{2} + \frac{2}{3} + \frac{xE[-2, -2] + yE[-2, -2, -1]}{xE[-2] + yE[-2, -1]}) = N(\frac{1}{2} + \frac{2}{3} + \frac{5x - 7y}{-2x + 3y})$. By theorem 5.4, $\frac{5x - 7y}{-2x + 3y} = \frac{3}{5} - 3 = \frac{-12}{5}$ which implies $\frac{x}{y} = -1$. Thus $U = (\frac{1}{2} + \frac{2}{3}) \circ (-2, -2, -1)$, $(\frac{2}{3} + \frac{1}{2}) \circ (-2, -2, -1)$ and $\frac{x}{y} = -1$.

(2) By theorem 5.18 (3b), $U = (\frac{-2}{1} * \frac{-3}{2}) \circ (-h_1, \cdots, -h_m, 0)$ or $(\frac{-3}{2} * \frac{-2}{1}) \circ (-h_1, \cdots, -h_m, 0)$ where $\frac{E[h_1, \cdots, h_{m-1}]}{E[h_2, \cdots, h_{m-1}]} = \frac{2}{3} + (-3) = \frac{-7}{3} = -2 + \frac{1}{-3} = (-3, -2)$. Thus $(h_{m-1}, \cdots, h_1) = (-3, -2)$. This implies that $m = 3$ and $(h_1, h_2, h_3) = (-2, -3, h_3)$ for $h_3 \leq 0$. Moreover, $N(\frac{1}{2} + \frac{3}{5} + \frac{2}{3} + (-3)) = N(\frac{1}{2} + \frac{2}{3} + \frac{yE[h_1, h_2] - xE[h_1, h_2, h_3]}{yE[h_2] - xE[h_2, h_3]}) = N(\frac{1}{2} + \frac{2}{3} + \frac{7y - x(7h_3 - 2)}{-3y - x(1 - 3h_3)})$. By theorem 5.4, $\frac{7y - x(7h_3 - 2)}{-3y - x(1 - 3h_3)} = \frac{3}{5} - 3 = \frac{-12}{5}$. Then $\frac{x}{y} = \frac{1}{h_3 - 2} = (1, h_3 - 3, 0)$ or $(-1, h_3 - 1, 0)$.

Hence, $U = (\frac{-2}{1} * \frac{-3}{2}) \circ (2,3,-h_3,0)$, $(\frac{-3}{2} * \frac{-2}{1}) \circ (2,3,-h_3,0)$ and $\frac{x}{y} = \frac{1}{h_3-2} = (1, h_3-3, 0)$ or $(-1, h_3-1, 0)$ for $h_3 \leq 0$.

Example 5.2. Solve

$$N(U + \frac{0}{1}) = N(\frac{7}{5})$$
$$\text{and} \quad N(U + \frac{x}{y}) = N(\frac{1}{2} + \frac{3}{5} + \frac{2}{3} - 3).$$

By theorem 5.19, $U = (\frac{c_1}{d_1} + \frac{7p-5c_1}{5d_1-7q}) \circ (h,0)$, $(\frac{7p-5c_1}{5d_1-7q} + \frac{c_1}{d_1}) \circ (h,0)$ and $N(\frac{c_1}{d_1} + \frac{7p-5c_1}{5d_1-7q} + \frac{x}{hx+y}) = N(\frac{1}{2}+\frac{3}{5}+\frac{2}{3}-3)$ for integers c_1, d_1, p and q such that $d_1p - qc_1 = 1$ and $0 < c_1 < d_1$. By theorem 5.4, $\frac{c_1}{d_1} = \frac{1}{2}, \frac{3}{5}$ or $\frac{2}{3}$.

(1) If $\frac{c_1}{d_1} = \frac{1}{2}$, then $\frac{7p-5c_1}{5d_1-7q} = \frac{7p-5}{10-7q} = \frac{2}{3} + k$ or $\frac{3}{5} + k$ for some integer k where $2p - q = 1$. Solving $10 - 7q = \pm 3$ and $2p - q = 1$ gives $p = q = 1$ and $k = 0$. However, there is no integers p, q such that $\frac{7p-5}{10-7q} = \frac{3+5k}{5}$ and $2p - q = 1$. Thus $U = (\frac{1}{2} + \frac{2}{3}) \circ (h,0)$ and $(\frac{2}{3} + \frac{1}{2}) \circ (h,0)$ and $\frac{x}{hx+y} = \frac{3}{5} - 3 - k = \frac{-12}{5}$. Thus $\frac{x}{y} = \frac{-12}{5+12h}$.

(2) If $\frac{c_1}{d_1} = \frac{3}{5}$, then $\frac{7p-5c_1}{5d_1-7q} = \frac{7p-15}{25-7q} = \frac{2}{3} + k$ or $\frac{1}{2} + k$ for some integer k where $5p - 3q = 1$. There is no such p and q.

(3) If $\frac{c_1}{d_1} = \frac{2}{3}$, then $\frac{7p-5c_1}{5d_1-7q} = \frac{7p-10}{15-7q} = \frac{1}{2} + k$ or $\frac{3}{5} + k$ for some integer k where $3p - 2q = 1$. There is no such p and q.

Hence, the solutions for this system of tangle equations are $U = (\frac{1}{2} + \frac{2}{3}) \circ (h,0)$ and $(\frac{2}{3} + \frac{1}{2}) \circ (h,0)$ and $\frac{x}{y} = \frac{-12}{5+12h} = (-2,-2,-2,-h,0)$ for any integer h.

5.4 Double branch covers

In this section, we discuss the correspondence between rational tangle replacement in tangle equations and surgery on a knot in the double branch cover of a substrate link. We assume a substrate link is either a Montesinos link or a rational link. The double branch cover of a rational link or a Montesinos link can be represented as a Seifert fiber space. Using this, theorem 5.23 shows that U is a generalized M-tangle iff the 0/1 tangle lifts to a fiber in a SFS with base surface S^2 for the double branch cover of M_1 in the system of tangle equations $N(U + \frac{0}{1}) = M_1$ and $N(U + \frac{x}{y}) = M_2$ where M_1 and M_2 are either Montesinos links or rational links.

Definition 5.8. [43] Let X and $\tilde{X}$ be path connected topological spaces. Then $(\tilde{X}, f)$ is called a *covering space* of X if $f : \tilde{X} \to X$ is a surjective continuous map with every $x \in X$ having an open neighborhood U such that each component C of $f^{-}(U)$ is open in $\tilde{X}$ and is homeomorphically mapped onto U by f.

Definition 5.9. [43] Let M and N be compact n-dimensional manifolds. Let $A \subset M$ and $B \subset N$ be proper submanifolds with codimension 2. Then a continuous map $f : M \to N$ is said to be a *branched covering* with *branch sets* A (upstairs) and B (downstairs) if

1. components of $f^{-1}(U_i)$ for open sets U_i of N are a basis for the topology of M,

2. $f(A) = B$ and $(M - A, f)$ is a covering space of $N - B$.

Each branch point $a \in A$ has a *branching index* k, meaning that f is k-to-one near a, not necessarily at a, and this number is constant on components of A. A standard example of a branched covering is $f : D^2 \to D^2$ defined by $f(z) = z^2$ where $D^2 = \{z \in \mathbb{C} | \|z\| = 1\}$. The branch point is the origin with index 2. For every $z \in \mathbb{C} - 0$, $f^{-1}(z)$ consists of two points. We say that f is a double branched covering branched at the origin.

All 3-manifolds and 2-manifolds which will be used in the rest of this chapter are assumed to be compact, connected and oriented. Dehn surgery on a knot in a 3-manifold M is a procedure which yields a new manifold by removing a regular neighborhood of a knot from M and gluing it back in a different way.

Definition 5.10. [43, 26] Let K be a knot in a manifold M, $N(K) \cong K \times D^2$ be a regular neighborhood of K, and $E_K = M - intN(K)$ be the exterior of K. Choose a meridian μ and a longitude λ on $\partial N(K)$ such that $i(\mu, \lambda) = \pm 1$. Then a *slope* on $\partial N(K)$ is an isotopy class of unoriented essential simple closed curves on $\partial N(K)$, $p \cdot \mu + q \cdot \lambda$ for some coprime integers p and q. α-*Dehn surgery* on a knot K is $K(\alpha) = K(\frac{p}{q}) = E_K \cup_h V$ where $h : \partial E_K \to \partial V$ is the orientation preserving homeomorphism such that $h(\alpha) = h(p \cdot \mu + q \cdot \lambda) =$ the meridian of the solid torus V. A simple example is $K(1/0) \cong S^3$ for all K in S^3.

As an extended notion of Dehn surgery, *Dehn filling* is a procedure which produces a new manifold from a manifold with a torus boundary component by filling the torus boundary with a solid torus along their boundaries as above. Thus Dehn surgery on a knot K corresponds to Dehn filling on E_K.

The double branch cover of a tangle T is denoted by $\widetilde{T}$. If T is a rational tangle, then $\widetilde{T}$ is a solid torus [36]. See Figure 5.9.

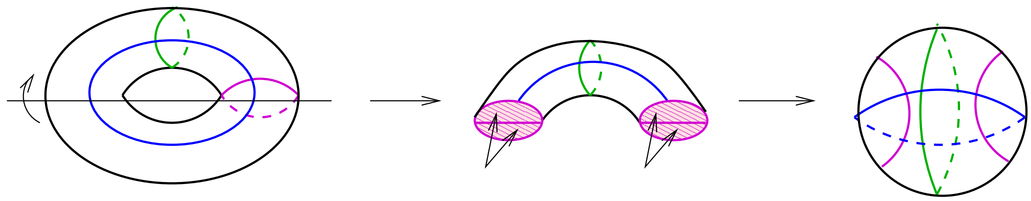


Figure 5.9: Double branch cover of a rational tangle.

Let us explain the relationship between rational tangle replacement and surgery on a knot in the double branch cover of a rational tangle [26]. Let V be a solid torus and $T^2 = \partial V$. Let $f : T^2 \rightarrow S^2$ be the double covering branched over $\{NW, NE, SW, SE\}$ on S^2. Let $h_n : B^3 \rightarrow B^3$ be a homeomorphism which sends 0-tangle to n-tangle and $v_n : B^3 \rightarrow B^3$ be a homeomorphism which sends ∞-tangle to 1/n-tangle. Let $\widetilde{h}_n$ and $\widetilde{v}_n : T^2 \rightarrow T^2$ be lifts of $h_n|S^2$ and $v_n|S^2$, respectively. That is, $f \circ \widetilde{h}_n = (h_n|S^2) \circ f$ and $f \circ \widetilde{v}_n = (v_n|S^2) \circ f$. Then $\widetilde{v}_n(m) = 1 \cdot m + n \cdot l$ where m and l are a meridian and a longitude of $\partial(\widetilde{\infty})$. Figure 5.10 shows an example when $n = 2$. Similarly, $\widetilde{h}_n(l) = n \cdot m + 1 \cdot l$. A meridian can be chosen uniquely up to ambient isotopy so that it bounds a disk in V and is essential in T^2. A longitude can be chosen so that it intersects with a meridian at a single point. There are infinitely many ambient isotopy classes of longitudes in T^2 [43]. If M and L are another choice of a meridian and a longitude in $\partial(\widetilde{\infty})$, then $\pm m \sim M$ and $\pm l \sim L + kM$ for some integer k. Thus $\widetilde{v}_n(M) = 1 \cdot M + n \cdot (L + kM) = (1 + nk) \cdot M + n \cdot L$ and $\widetilde{h}_n(L) = n \cdot M + 1 \cdot (L + kM) = (n + k) \cdot M + 1 \cdot L$.

The homeomorphism $\widetilde{v}_n$ is represented as the matrix $\left(\begin{smallmatrix} 1 & 0 \\ n & 1 \end{smallmatrix}\right)$ and $\widetilde{h}_n$ is represented as the matrix $\left(\begin{smallmatrix} 1 & n \\ 0 & 1 \end{smallmatrix}\right)$. Every rational tangle can be written as a vector of even length [19]. Let $p/q = (x_1, \cdots, x_{2k})$ be a rational tangle. Then the corresponding boundary homeomorphism $g : \partial(\widetilde{p/q}) \to \partial(\widetilde{\infty})$ is represented as a matrix as follows [19]:

$$\begin{pmatrix} 1 & x_{2k} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ x_{2k-1} & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & x_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ x_1 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} E[x_1, \cdots, x_n] & E[x_2, \cdots, x_n] \\ E[x_1, \cdots, x_{n-1}] & E[x_2, \cdots, x_{n-1}] \end{pmatrix} = \begin{pmatrix} p & p' \\ q & q' \end{pmatrix} \text{ where } pq' - qp' = 1.$$

Thus for a meridian M and a longitude L of $\partial(\widetilde{\infty})$, replacement of the ∞ tangle with p/q tangle corresponds to a homeomorphism sending M to $p \cdot M + q \cdot (L + kM) = (p + kq) \cdot M + q \cdot L$ for some integer k. That is, replacement of the ∞ tangle with p/q tangle corresponds to $(p + kq)/q$ surgery on a knot in the double branch cover of the ∞ tangle.

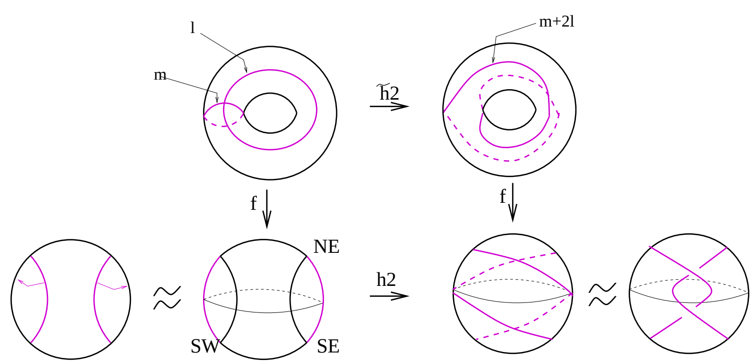


Figure 5.10: The corresponding homeomorphisms (upstairs) to vertical twists (downstairs).

Definition 5.11. [47] A *Seifert fiber space* (SFS) is a 3-manifold N which can be decomposed as the disjoint union of circles (fibers). Each fiber has a *fibered solid torus neighborhood* which is obtained from $D^2 \times I$ by rotating the top $D^2 \times \{1\}$ by $r = 2\pi a/b$ with the bottom $D^2 \times \{0\}$ fixed and identifying the top and the bottom where $b \geq 1, (a, b) = 1$. The central fiber is called an *exceptional fiber* of multiplicity b if $b > 1$. Otherwise, it is called an *ordinary fiber*. Each fiber of the fibered solid torus neighborhood except for the central fiber is an ordinary fiber and consists of the union of b segments $x \times I, r(x) \times I, \cdots, r^{b-1}(x) \times I$ for $x \in D^2 \backslash (0, 0)$.

Figure 5.11 shows an example of a fibered solid torus neighborhood when $a = 1$ and $b = 3$. Three blue segments are connected to become an ordinary fiber and the red segment becomes an exceptional fiber of multiplicity 3. The quotient space of a SFS N obtained by identifying each fiber to a point is called the *base (orbit) surface* B of N.

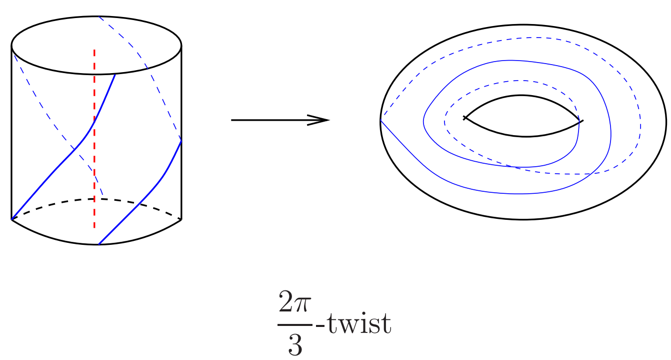


Figure 5.11: Fibered solid torus neighborhood.

Let N be a SFS with base surface B and n exceptional fibers of multiplicities $q_1, \cdots, q_n$. Let V_i be a regular neighborhood of the ith exceptional fiber. Let $N_0 = N - int \coprod_{i=1}^{n} V_i \cong B_0 \times S^1$ where $B_0 = B - n$ open disks. Let $Q_i = B_0 \cap \partial V_i$ and $H_i = \{1\} \times S^1$ be a basis for $H_1(\partial V_i)$. Q_i is called a *crossing curve* and H_i is a *fiber* in ∂N_0. Then N can be constructed as $N = N_0 \cup_{i=1}^{n} W_i$ where W_i's are solid tori and the meridian M_i of ∂W_i is sent to $q_i Q_i + p_i H_i$ in $\partial V_i \subset \partial N_0$ for some p_i which is coprime with q_i for $1 \leq i \leq n$. Then we write $N = B(\frac{p_1}{q_1}, \cdots, \frac{p_n}{q_n})$ where $q_i \geq 0$. It is called a *generalized SFS* if $q_i = 0$ for some i. Choose a longitude L_i of ∂W_i so that $\{M_i, L_i\}$ is a basis for $H_1(\partial W_i)$. Then $L_i = q_i' Q_i + p_i' H_i$ for q_i', p_i' such that $det \left(\begin{smallmatrix} q_i & p_i \\ q_i' & p_i' \end{smallmatrix}\right) = 1$. If we solve for Q_i and H_i in terms of M_i and L_i, then we get $Q_i = p_i' M_i - p_i L_i$ and $H_i = -q_i' M_i + q_i L_i$. Since M_i is homologically trivial in W_i, $Q_i = -p_i L_i$ and $H_i = q_i L_i$ in $H_1(W_i)$. That is $-p_i$ is the number of times Q_i wraps around W_i and q_i is the number of times H_i wraps around W_i [30]. That is, H_i has multiplicity q_i.

Theorem 5.21. *[30] Two Seifert fiber spaces $B(\frac{p_1}{q_1}, \cdots, \frac{p_n}{q_n})$ and $B'(\frac{p_1'}{q_1'}, \cdots, \frac{p_m'}{q_m'})$ are fiber preserving homeomorphic or equivalent iff $B = B'$ and $B'(\frac{p_1'}{q_1'}, \cdots, \frac{p_m'}{q_m'})$ can be obtained from $B(\frac{p_1}{q_1}, \cdots, \frac{p_n}{q_n})$ by the following collection of operations:*

1. *Add or delete $\frac{0}{1}$,*
2. *Replace $\frac{\pm 1}{0}$ by $\frac{\mp 1}{0}$,*
3. *Replace $\frac{p_i}{q_i}$ by $\frac{p_i}{q_i} + K_i$ where $\sum K_i = 0$.*

For example, $B(\frac{1}{2}, \frac{-3}{2}) = B(\frac{1}{2}, \frac{-3}{2}, \frac{0}{1}) = B(\frac{-1}{2}, \frac{-1}{2}, \frac{0}{1})$.

Now, let us explore double branch covers of rational tangles (links) and Montesinos tangles (links) as Seifert fiber spaces. The double branch cover of a rational tangle as a SFS has a base surface D^2 and 0 or 1 exceptional fiber. There are many choices of fibrations. Here, we choose a longitude of the solid torus to be a fiber so that $\widetilde{0}$ is $D^2(0/1)$. Similarly, a meridian of the solid torus can be chosen to be a fiber so that $\widetilde{\infty}$ is $D^2(1/0)$. In general, $D^2(p/q)$ can be chosen to be a SFS for $\widetilde{p/q}$. The double branch cover of the sum of two rational tangles $\frac{p_1}{q_1}$ and $\frac{p_2}{q_2}$ is the SFS $D^2(\frac{p_1}{q_1}, \frac{p_2}{q_2})$. Considering $\frac{p_1}{q_1}$ as $\frac{p_1}{q_1} + \frac{0}{1}$, replacement of $\frac{0}{1}$ with $\frac{p_2}{q_2}$ corresponds to drilling out a small solid torus from the double branch cover $D^2(\frac{p_1}{q_1})$ of $\frac{p_1}{q_1}$ and gluing a solid torus by identifying $q_2Q + p_2H$ to the meridian of the gluing solid torus assuming a longitude of $\partial(D^2(\frac{p_1}{q_1}) -$ the small sold torus removed) is the fiber H. This results in the SFS $D^2(\frac{p_1}{q_1}, \frac{p_2}{q_2})$. See Figure 5.12. Generally, the double branch cover of $\frac{p_1}{q_1} + \cdots + \frac{p_n}{q_n}$ is the SFS $D^2(\frac{p_1}{q_1}, \cdots, \frac{p_n}{q_n})$.

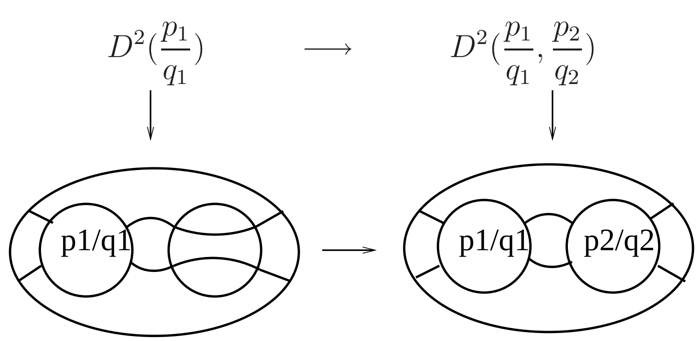


Figure 5.12: Seifert fiber space of sum of two tangles.

The numerator closure of a tangle $\frac{p}{q}$, $N(\frac{p}{q})$ can be considered as the union of the tangle $\frac{p}{q}$ and the 0-tangle glued along their boundaries by identity map id_∂ on the boundary. That is, $N(\frac{p}{q}) = \frac{p}{q} \cup_{id_\partial} \frac{0}{1}$. Thus the double branch cover of $N(\frac{p}{q})$, $\widetilde{N(\frac{p}{q})}$ is $\frac{\widetilde{p}}{q} \cup_{id_\partial} \frac{\widetilde{0}}{1} = D^2(\frac{p}{q}) \cup_{id_\partial} D^2(\frac{0}{1})$ which is the SFS with base surface $D^2 \cup_{id_\partial} D^2$ with one exceptional fiber. Thus a Seifert fibration of the double branch cover of $N(\frac{p}{q})$ is $S^2(\frac{p}{q})$. Generally, the numerator closure $N(\frac{p_1}{q_1} + \cdots + \frac{p_n}{q_n})$ has the double branch cover $S^2(\frac{p_1}{q_1}, \cdots, \frac{p_n}{q_n})$.

SFS is not unique for a lens space which is the double branch cover of a rational link [30].

Theorem 5.22. *[30] The double branch cover of a Montesinos link has a unique SFS up to fiber preserving homeomorphism except for the following two cases:*

1. $RP^2(\frac{\beta}{\alpha}) \cong S^2(\frac{1}{2}, \frac{-1}{2}, \frac{\alpha}{-\beta})$ *where RP^2 is the real projective plane.*
2. $RP^2 \sharp RP^2(\frac{0}{1}) \cong S^2(\frac{1}{2}, \frac{1}{2}, \frac{-1}{2}, \frac{-1}{2})$.

Note that two SFS in 1 and 2 are homeomorphic up to orientation preserving, but not fiber preserving, homeomorphism.

Theorem 5.23. *Let $N(U + \frac{0}{1}) = N(\frac{a_1}{b_1} + \cdots + \frac{a_s}{b_s} + e_1)$ and $N(U + \frac{x}{y}) = N(\frac{z_1}{v_1} + \cdots + \frac{z_t}{v_t} + e_2)$. Then U is a generalized M-tangle iff there exists a generalized SFS for the double branch cover of the substrate link such that the generalized SFS has base surface S^2 and the $0/1$ tangle lifts to a neighborhood of a fiber and $\widetilde{U}$ is a SFS.*

Proof. $N(U + \frac{0}{1}) = U \cup_{id_\partial} \frac{0}{1}$ where $id_\partial : \partial(U) \to \partial(\frac{0}{1})$ is the identity map.

($\Rightarrow$) If U is a generalized M-tangle, then $\widetilde{U}$ can be chosen to be a SFS with base surface D^2. Then the fibration of $\widetilde{U}$ can extend to $\widetilde{\frac{0}{1}}$ to get a generalized SFS for the double branch cover of the substrate link. Thus $\widetilde{\frac{0}{1}}$ is a SFS with base surface D^2 and $0/1$ lifts to a neighborhood of a fiber in the generalized SFS with base surface S^2.

($\Leftarrow$) Suppose that $0/1$ lifts to a neighborhood of a fiber in a generalized SFS with base surface S^2 for the double branch cover of the substrate knot. Since $\widetilde{U} \cup_{\widetilde{id_\partial}} \widetilde{\frac{0}{1}}$ where $\widetilde{id_\partial}$ is the lift of id_∂, is a generalized SFS with base surface S^2 and $\widetilde{\frac{0}{1}}$ is a neighborhood of a fiber, $\widetilde{U} = (\widetilde{U} \cup_{\widetilde{id_\partial}} \widetilde{\frac{0}{1}}) - \widetilde{\frac{0}{1}}$ is a SFS with base surface D^2. By theorem 8 in [19], U is a generalized M-tangle. $\square$

CHAPTER 6
CONCLUSION AND FUTURE DIRECTION

Completing a knot distance table for knots up to 10 crossings is a long-term goal. Knot distances can be calculated using mathematical theories and computer simulations. Knot distances are related to the study of topoisomerase action and the study of knot theory such as conjectures on unknotting numbers. Thus calculation of knot distances is interesting both biologically and mathematically. Knot distances were calculated using some mathematical theories by Isabel K. Darcy [10, 18]. In my thesis, undetermined knot distances in the old knot distance table by Isabel K. Darcy are improved based on the Jones polynomial.

We generalized A. Stoimenow's theory about the unknotting number in [48] to the signed knot distance cases in theorems 4.1 and 4.2. Theorems 4.1 and 4.2 show that if a signed knot distance, $d_{++}(K_1, K_2)$ or $d_{--}(K_1, K_2)$, between two knots K_1 and K_2 is equal to one, then there exists a knot K' obtained by horizontal smoothing of the crossing of K_1 where the crossing change occurs such that the Jones polynomial of K' is obtained using the Jones polynomials of K_1 and K_2. If we consider L' in theorems 4.1 and 4.2 obtained via the vertical smoothing of the crossing of K_1 where the crossing change occurs, then L' is a two component link. Thus we expect to discover more new lower bounds of signed knot distances if the Jones polynomial of links and its properties are involved.

In my thesis, we improve knot distances by combining two signed knot distances, $d_{++}(K_1, K_2)$ and $d_{--}(K_1, K_2)$. That is, if $d_{++}(K_1, K_2) \geq 2$ and $d_{--}(K_1, K_2) \geq$

2, then $d(K_1, K_2) \geq 2$. The data obtained from our algorithm calculating knot distances can be directly used to improve signed knot distances. We also addressed the possibility that d_{++} and d_{--} can be improved to 2 by separately using different theories. For this purpose, the signature of knots is employed in section 4.4 to improve signed knot distances which were not improved based on the Jones polynomial. Even though we did not detect any improvement by checking the signature conditions in lemmas 4.3 and 4.3 after utilizing the Jones polynomial, this idea gives us a future direction to improve knot distances based on signed knot distances which can be more efficient than improving knot distances themselves. In addition, theories about unknotting numbers can be generalized to calculate knot distances since the unknotting number is a special case of the knot distance. There are many theories which determine some unknotting numbers [27, 40, 39].

There are also other types of distances between knots. For example, a $(0, 2)$-move in section 5.2 corresponds to a crossing change and a $(\pm 1, \infty)$-move corresponds to smoothing a crossing. Since these moves are related to the study of various enzyme mechanisms, calculating other types of distances is also interesting biologically as well as mathematically. Theorems 4.1 and 4.2 suggest that the Jones polynomial can be used to improve lower bounds of smoothing distances which are related to smoothing a crossing.

We are also interested in improving upper bounds as well as lower bounds of knot distances. Shorter pathways between two knots via crossing changes may be found if at least one of them is a Montesinos link by using theorems 5.18, 5.19 and

5.20. In section 5.3, systems of tangle equations $N(U+\frac{0}{1}) = K_1$ and $N(U+\frac{x}{y}) = K_2$ are solved for U where $\frac{x}{y}$ is a rational tangle, U is a generalized Montesinos tangle and K_1 and K_2 are either Montesinos links or rational links. We assume that at least one of K_1 and K_2 is a Montesinos link. Thus solving these system of tangle equations involving Montesinos links when $\frac{x}{y}$ is the 2-tangle can improve upper bounds of the crossing change knot distance. Solving tangle equations involving Montesinos links can also be utilized in studying other types of knot distances and thus various enzyme mechanisms.

We also plan to implement solutions of tangle equations in section 5.3 to 'Topoice' which is a software modeling enzyme actions in 'KnotPlot'. Tangle equations when K_1 and K_2 are both rational knots were solved by Isabel K. Darcy and the solutions were implemented in 'Topoice'. By implementing solutions of tangle equations involving Montesinos links, we can solve more tangle equations modeling enzyme actions.

APPENDIX A
REAME FILE

This is the README file of programs for 1)generating the Jones polynomials of rational knots based on the program for calculating HOMFLYPT polynomial by Bruce Ewing and Ken Millett, 2)checking the properties of the Jones polynomial in section 4.2 to determine whether a given polynomial could be the Jones polynomial of a knot and 3)generating a knot distance table. Two programs written by me are included in appendix B: EM.c and Jones_2.m. The other programs for generating the Jones polynomials were modified from Bruce Ewing and Ken Millett's program and the programs for generating a knot distance table were modified from Isabel K. Darcy's programs. They are not included in my thesis. All programs run on Linux.

A.1 Generating the Jones polynomial

The following programs are used to generate the Jones polynomial of rational knots up to 13 crossings: EM.c, lmpoly02.c, and convsym_Jones2.c. EM.c is a C++ code which expands a rational number into a vector whose entries are all even integers based on a modified division algorithm and creates EM codes for rational knots. EM.c is included in appendix B. The program lmpoly02.c generates a matrix form of HOMFLYPT polynomial using an EM code. This matrix form of HOMFLYPT polynomial is converted to the Jones polynomial by convsym_Jones2.c. The program lmpoly02.c is B. Ewing and K. Millett's code and their program 'convsym.c' is modified to 'convsym_Jones2.c' by substituting two variables l and m with it and

$i(t^{-\frac{1}{2}} - t^{\frac{1}{2}})$, respectively.

A.1.1 EM.c

EM.c has 'main' followed by two subroutines, 'FractionExpand_2' and 'EwingMillett'. In main, the integers α and β are initialized for rational knots up to 13 crossings by knots_1.c where $S(\alpha, \beta)$ represents a rational knot. They are used to print out the knot name corresponding to $S(\alpha, \beta)$ into a file using KnotOrder.c and KnotOrder2.c and obtain the corresponding continued fraction expansion whose entries are all even integers using EM.c. 'FractionExpand_2' expands a rational number into a vector of all even integers and 'EwingMillett' generates an EM code based on a vector obtained by 'FractionExpand_2.'

To compile EM.c, type 'g++ -c EM.c knots_1.c KnotOrder.c KnotOrder2.c' and then type 'g++ -o em EM.o knots_1.o KnotOrder.o KnotOrder2.o'. This produces an executable file 'em'. To run it, type './em'. This creates an output file 'EMTable' which has the list of EM codes for rational knots up to 13 crossings. The programs knots_1.c, KnotOrder.c and KnotOrder2.c are based on Isabel K. Darcy's knot distance programs. The programs knots_1.c and KnotOrder2.c were modified to re-index knot orders to include nonrational prime 9 crossing knots.

A.1.2 lmpoly02.c

The program lmpoly02.c is Bruce Ewing and Ken Millett's program for generating a matrix form of HOMFLYPT polynomial using EM code. This readme file is written based on their readme file. To compile, type 'gcc lmpoly02.c -o lmpoly02' where lmpoly02 is an executable file. To run it, type './lmpoly02 EMTable' where EMTable is a file containing the list of EM codes. The output is written in 'lm-knot.out'. This file is overwritten whenever it runs with a file containing EM codes. In this file, a HOMFLYPT polynomial is represented as a two dimensional matrix of coefficients for the powers of the two variables l and m. An example below is the matrix form of the HOMFLYPT polynomial of the trefoil knot:

$$trefoil$$

$$[[0]\,0\ -2\,0\ -1]$$

$$[0]$$

$$[0]\,0\,1$$

The powers of l are written horizontally with a pair of square brackets surrounding the zero power and with negative powers written to the left of the zero power. The powers of m are written vertically with a pair of square brackets surrounding the row containing the zero power of m and the most negative power is written at the top. The above matrix can be converted into the HOMFLYPT polynomial:

$$(0 \cdot l^0 + 0 \cdot l^1 - 2 \cdot l^2 + 0 \cdot l^3 - 1 \cdot l^4) \cdot m^0 + (0 \cdot l^0)m^1 + (0 \cdot l^0 + 0 \cdot l^1 + 1 \cdot l^2)m^2$$

$$= -2l^2 - l^4 + l^2 m^2.$$

A.1.3 convsym_Jones2.c

The program convsym_Jones2.c was modified from Bruce Ewing and Ken Millett's program for converting a matrix form of HOMFLYPT polynomial to a HOM-

FLYPT polynomial by substituting two variables l and m with it and $i(t^{-\frac{1}{2}} - t^{\frac{1}{2}})$, respectively. To compile, type 'gcc convsym_Jones2.c -o convsym_J2' where convsym_J2 is an executable file. To run it, type './convsym_J2 lmknot.out JonesP' where lmknot.out is where matrix HOMFLYPT polynomials are stored and JonesP is the file where the Jones polynomials are written. The Jones polynomials in the output file are not simplified. They can be simplified using other programs such as Matlab or Mathematica.

A.2 Checking the properties of the Jones polynomial

Jones_2.m is code written in MATLAB to check the properties of the Jones polynomial in section 4.2 for the polynomials in theorems 4.1 and 4.2 which are calculated based on pairs of knots whose distance have lower bound one. It also checks the signature conditions in lemmas 4.3 and 4.4. The list of pairs of knots with lower bound one (pair.txt), the list of the Jones polynomial of rational knots up to 13 crossings and nonrational prime 8 and 9 crossing knots (JonePoly13_new.txt) and the list of signatures for rational knots up to 13 crossings and nonrational 8 crossing knots (Signature.txt) are imported.

A.3 Generating a knot distance table

The programs for calculating and tabulating knot distances by Isabel K. Darcy were revised to generate a new knot distance table. The knot distance program 'knotdistance.tar.gz' is available upon request. It includes the following files:

- Makefile, Change_Sign.c, NewSign2.c, knots_1.c, knot9.c, KnotOrder.c, KnotOrder2.c, DistanceoneNon4plats.c, Mathematics2.c, Distanceone.c, Theorem1a.c, Mirror.c, ConnectedSum.c,Lowerbounds.c, DistTriangle2.c, SignTriangle2.c, Tablf.c, metric4.c

The knot distance program is compiled by typing 'make -f Makefile'. It creates an executable file 'general2'. New data obtained by the Jones polynomial and information such as alpha and beta for nonrational prime 9 crossing knots are included to generate a new knot distance table. Alpha for a nonrational prime knot is assigned to be -3 and beta is the determinant of the knot. We also re-indexed knot orders to add nonrational prime 9 crossing knots. New knot distances are colored red in the new knot distance table.

Change_Sign.c and NewSign2.c are used to implement the new data obtained by the Jones polynomial. The program knot9.c assigns information such as alpha an beta for nonrational prime 9 crossing knots. In programs knot_1.c, KnotOrder2.c and metric4.c, knot numbering was re-indexed to include nonrational prime 9 crossing knots into a knot distance table.

APPENDIX B
PROGRAM FILES

B.1 EM.c

```
/*This program generates EM codes for rational knots.
It calls the programs knots_1.c, KnotOrder.c and KnotOrder2.c.
The program knots_1.c initializes knot information such as
alpha and beta. KnotOrder.c and KnotOrder2.c assign Rolfsen
knot names based on alpha and beta for rational knots and
based on the knot numbering for nonrational prime 8 and
9 crossing knots, respectively.*/

/*This program has two subroutines in addition to 'main':
'FractionExpand_2' expands a rational number into a vector of
all even integers and 'EwingMillett' generates an EM code
based on a vector obtained by 'FractionExpand_2.'*/

#include <stdlib.h>
#include <stdio.h>
#include <assert.h>
#include <math.h>
#include <errno.h>

#include "var.h"
#include "knot.h" /*used for assigning alpha and beta*/
#include "KnotOrder.h" /*used for assigning knot names*/

int FractionExpand_2(int, int, int []);

void EwingMillett(int [], int, FILE*);

int main()
{
  int alpha1,beta1,alpha2,beta2,alpha3,beta3,k1,k2,k3,i;
  int c1[100]={0};
  int c3[100]={0};
  int c4[100]={0};
  int s1, s2, s3;

  int *a;
  int *b;
```

```
  int *c;
  int *c2;
  int *g;

  a = (int *) calloc (MAX, sizeof (int));
  b = (int *) calloc (MAX, sizeof (int));
  c = (int *) calloc (MAX, sizeof (int));
  c2 = (int *) calloc (MAX, sizeof (int));
  g = (int *) calloc (MAX, sizeof (int));

  /*initialize the knot information in knots_1.c*/
  init_globals_1();

  FILE *tabl;
  tabl = fopen("EMTable","w");

 for(i=0;i<MAX;i++)
{
  if(alpha[i]>0)
  {
    /*assign the knot name*/
     fprintf(tabl,"#%d (*",i);
     KnotOrder(tabl,beta[i], alpha[i],i);
     KnotOrder2(tabl,i);
     fprintf(tabl,"*)");
     fprintf(tabl,"\n");

    /*print out the EM code into a file*/
    alpha1=beta[i];
    beta1=alpha[i];
    k1= FractionExpand_2(alpha1,beta1,c1);
    EwingMillett(c1,k1,tabl);
    fprintf(tabl,"\n");
  }
 }
 fprintf(tabl,"%end\n");
 fclose(tabl);
 return 0;
} /*end of main*/

/*  FractionExpand: Given alpha and beta, finds the continued
fraction expansion < c[0], ..., c[k-1] > = s(alpha,beta)
where all c[i] are nonzero even integers and also returns
k = length */

int FractionExpand_2( int alpha, int beta, int c[] )
```

```
{
  int k;
  int r[20];

 if(beta%2==1 || beta%2==-1)
  {
       if(alpha%2==1 || alpha%2==-1)
       beta = beta+alpha;
  }
 if ( beta < 0 && alpha < 0 )
  {
    beta = -beta;
    alpha = - alpha;
  }
  k = 1;
  if ( alpha == 0 )
        c[0] = 0;
  else if ( beta == 0 )
       {
        c[0] = 2;
        c[1] = 0;
      k = 2;
       }
  else
  {
      c[0] = alpha/beta;
      if(c[0]%2==1)
      {
        c[0]=c[0]+1;
        r[1]=(alpha%beta)-beta;
        r[0]=beta;
      }
      else if(c[0]%2==-1)
      {
        c[0]=c[0]-1;
        r[1]=(alpha%beta)+beta;
        r[0]=beta;
      }
      else
      {
            r[1] = alpha%beta;
            r[0] = beta;
      }
      while ( r[k] != 0 )
        {
             c[k] = r[k-1]/r[k];
```

```
                 if(c[k]%2==1)
                {
                    c[k]=c[k]+1;
                    r[k+1]=(r[k-1]%r[k])-r[k];
                }
                else if(c[k]%2==-1)
                {
                    c[k]=c[k]-1;
                    r[k+1]=(r[k-1]%r[k])+r[k];
                }
                else
                {
                    r[k+1] = r[k-1]%r[k];
                }
                    k = k + 1;
            }
      }

/*remove all zeros in the continued fraction <c[0],..,c[k-1]>*/

int i,j;
int d[20];

for(i=1;i<k-1;i++)
{
   if(c[i]==0)
    {
       k=k-2;
       d[i-1]=c[i-1]+c[i+1];
       for(j=i;j<k;j++)
       {
        d[j]=c[j+2];
       }
       for(j=i-1;j<k;j++)
       {
        c[j]=d[j];
       }
    }
}

if(c[0]==0)
{
  if(k>2)
   {
    k=k-2;
    for(j=0;j<k;j++)
```

```
     {
       c[j]=c[j+2];
     }
    }
  else if(k==2) /*unknot*/
  {
       c[1]=2;
  }
}

if(c[k-1]==0)
{
  if(k>2)
   {
    k=k-2;
   }
  else if(k==2) /*unknot*/
  {
      c[0]=2;
  }
}
return (int) k;
/* length = k  */
}


/* print out the Ewing Millett code for a rational knot with
continued fraction <c[0],...,c[k-1]> of all even integers */

void EwingMillett(int c[], int k, FILE* f)
{
  int i,j;
  int l=0;
  int ca[k];  /*ca[k]==|c[k]|*/

  for(i=0;i<k;i++)
  {
    if(c[i]≥0)
    {
    ca[i]=c[i];
    }
    else if(c[i]≤0)
    {
    ca[i]=-c[i];
    }
  }
```

```
  for(i=0;i<k;i++)
    l += ca[i]; /*l = crossing #*/

  int a[5*l];
  char b[5*l+1];
  b[5*l+1]='\0';

  int *ap;
  char *bp;
  ap=a;
  bp=b;

/*assign numbers and signs to each crossing
 in <c[0],c[1],...,c[k-2],c[k-1]>*/

  for(i=0;i<l;i++)
  {
    *ap=i+1;
    ap += 5;
  }
  ap -= 5*l;

  for(j=0;j<k;j++)
  {
    for(i=0;i<ca[j];i++)
    {
    if(c[j]>0)
    {
      if(j%2==0)
        *bp='+';
      else
        *bp='-';
    }
    else if(c[j]<0)
    {
      if(j%2==0)
        *bp='-';
      else
        *bp='+';
    }
    else
    {
      printf("mistake \n");
    }
    bp += 5;
```

```
    }
  }
  bp -= 5*l;

/*1)-keep track of codes from C[0] to c[k-1]
in <c[0],c[1],...,c[k-2],c[k-1]>*/

int d=1;
if(c[0]>0)
{
  ap += 4;
  bp += 4;
  *ap = ++d;
  *bp = 'c';
  ap += 2;
  bp +=2;
  while(ca[0]>d)
  {
     *ap = ++d;
     *bp = 'b';
     ap += 8;
     bp += 8;
     *ap = ++d;
     *bp = 'c';
     ap += 2;
     bp +=2;
  }
}
else if(c[0]<0)
{
  ap += 1;
  bp += 1;
  *ap = ++d;
  *bp = 'd';
  ap += 6;
  bp += 6;
  while(ca[0]>d)
  {
     *ap = ++d;
     *bp = 'c';
     ap += 4;
     bp += 4;
     *ap = ++d;
     *bp = 'd';
     ap += 6;
     bp += 6;
```

```
  }
}

int s=ca[0];
for(i=1;i<k;i++)
{
  s += ca[i];
  /*case 1*/
  if(c[i]>0 && i%2==1)
  {
       *ap = ++d;
       *bp = 'd';
       if(c[i-1]>0)
       {
        ap +=6;
        bp +=6;
       }
       else if(c[i-1]<0)
       {
        ap +=5;
        bp +=5;
       }
       *ap = ++d;
       *bp = 'c';
        ap += 4;
        bp += 4;
       while(s>d)
       {
       *ap = ++d;
       *bp ='d';
        ap +=6;
        bp +=6;
       *ap = ++d;
       *bp ='c';
        ap +=4;
        bp +=4;
       }
  }
  /*case 2*/
  else if(c[i]<0 && i%2==1)
  {
       *ap = ++d;
       *bp = 'c';
       if(c[i-1]>0)
       {
         ap +=5;
```

```
    bp +=5;
  }
  else if(c[i-1]<0)
  {
    ap +=4;
    bp +=4;
  }
  *ap = ++d;
  *bp = 'b';
   ap += 8;
   bp += 8;
  while(s>d)
  {
  *ap = ++d;
  *bp ='c';
   ap +=2;
   bp +=2;
  *ap = ++d;
  *bp ='b';
   ap +=8;
   bp +=8;
  }
}
/*case 3*/
else if(c[i]>0 && i%2==0)
{
  *ap = ++d;
  *bp = 'b';
  if(c[i-1]>0)
  {
    ap +=8;
    bp +=8;
  }
  else if(c[i-1]<0)
  {
   ap +=5;
   bp +=5;
  }
  *ap = ++d;
  *bp = 'c';
   ap += 2;
   bp += 2;
 while(s>d)
  {
   *ap = ++d;
   *bp ='b';
```

```
         ap +=8;
         bp +=8;
        *ap = ++d;
        *bp ='c';
        ap +=2;
        bp +=2;
       }
 }
  /*case 4*/
  else if(c[i]<0 && i%2==0)
  {
       *ap = ++d;
       *bp = 'c';
       if(c[i-1]>0)
       {
        ap +=5;
        bp +=5;
       }
       else if(c[i-1]<0)
       {
        ap +=2;
        bp +=2;
       }
       *ap = ++d;
       *bp = 'd';
        ap += 6;
        bp += 6;
     while(s>d)
       {
        *ap = ++d;
        *bp ='c';
         ap +=4;
         bp +=4;
        *ap = ++d;
        *bp ='d';
         ap +=6;
         bp +=6;
       }
  }
} /*for*/

/*2) keep track of c[k-1] to c[k-2]
in <c[0],c[1],...,c[k-2],c[k-1]>*/
s = l-ca[k-1]-ca[k-2]+1;
if(c[k-2]>0)
{
```

```
     d -= ca[k-1];
    *ap = d;
    *bp = 'b';
    if(c[k-1]>0)
    {
      ap -= 2+5*(ca[k-1]-1);
      bp -= 2+5*(ca[k-1]-1);
    }
    else if(c[k-1]<0)
    {
      ap -= 5*ca[k-1];
      bp -= 5*ca[k-1];
    }
    *ap = --d;
    *bp = 'c';
     ap -= 8;
     bp -= 8;

     while(s<d)
     {
       *ap = --d;
       *bp = 'b';
       ap -= 2;
       bp -= 2;
       *ap = --d;
       *bp = 'c';
       ap -= 8;
       bp -= 8;
     }
  }  /*if*/

  else if(c[k-2]<0)
  {
    d -= ca[k-1];
    *ap = d;
    *bp = 'c';
    if(c[k-1]>0)
    {
      ap -= 5*ca[k-1];
      bp -= 5*ca[k-1];
    }
    else if(c[k-1]<0)
    {
      ap -= 8+5*(ca[k-1]-1);
      bp -= 8+5*(ca[k-1]-1);
    }
```

```
  *ap = --d;
  *bp = 'd';
   ap -= 4;
   bp -= 4;

   while(s<d)
   {
     *ap = --d;
     *bp = 'c';
     ap -= 6;
     bp -= 6;
     *ap = --d;
     *bp = 'd';
     ap -= 4;
     bp -= 4;
   }
}

/*3) c[k-2]-->c[k-4]-->...-->c[2]-->c[0]
in <c[0],c[1],...,c[k-2],c[k-1]>*/
i = k-2;
s = l-ca[k-1]-ca[k-2];
while(i>0)
{
  s -= ca[i-1]+ca[i-2];
  if(c[i-2]>0)
  {
    d -= ca[i-1]+1;
    *ap = d;
    *bp = 'b';
    if(c[i]>0)
    {
      ap -= 2+5*ca[i-1];
      bp -= 2+5*ca[i-1];
    }
    else if(c[i]<0)
    {
      ap -= 3+5*ca[i-1];
      bp -= 3+5*ca[i-1];
    }
    *ap = --d;
    *bp = 'c';
    ap -= 8;
    bp -= 8;

    while(s<d-1)
```

```
      {
        *ap = --d;
        *bp = 'b';
        ap -= 2;
        bp -= 2;
        *ap = --d;
        *bp = 'c';
        ap -= 8;
        bp -= 8;
      }
    } /*if*/

   else if(c[i-2]<0)
    {
      d -= ca[i-1]+1;
      *ap = d;
      *bp = 'c';
      if(c[i]>0)
      {
        ap -= 5+5*ca[i-1];
        bp -= 5+5*ca[i-1];
      }
      else if(c[i]<0)
      {
        ap -= 6+5*ca[i-1];
        bp -= 6+5*ca[i-1];
      }
      *ap = --d;
      *bp = 'd';
      ap -= 4;
      bp -= 4;

      while(s<d-1)
      {
        *ap = --d;
        *bp = 'c';
        ap -= 6;
        bp -= 6;
        *ap = --d;
        *bp = 'd';
        ap -= 4;
        bp -= 4;
      }
    }/*else if*/
    i -= 2;
  } /*while*/
```

```
/*4) keep track of c[0] to c[k-1]
in <c[0],c[1],...,c[k-2],c[k-1]>*/
s=l-ca[k-1]+1;
if(c[k-1]>0)
{
  d += l-1;
  *ap = d;
  *bp = 'd';
  if(c[0]>0)
  {
    ap += 1+5*(l-1);
    bp += 1+5*(l-1);
  }
  else if(c[0]<0)
  {
    ap += 5*(l-1);
    bp += 5*(l-1);
  }
  *ap = --d;
  *bp = 'c';
  ap -= 6;
  bp -= 6;

  while(s<d)
  {
     *ap = --d;
     *bp = 'd';
     ap -= 4;
     bp -= 4;
     *ap = --d;
     *bp = 'c';
     ap -= 6;
     bp -= 6;
  }
} /*if*/

else if(c[k-1]<0)
{
  d += l-1;
  *ap = d;
  *bp = 'c';
  if(c[0]>0)
  {
    ap += 5*(l-1);
    bp += 5*(l-1);
```

```
    }
    else if(c[0]<0)
    {
      ap += -1+5*(l-1);
      bp += -1+5*(l-1);
    }
    *ap = --d;
    *bp = 'b';
    ap -= 2;
    bp -= 2;

    while(s<d)
    {
       *ap = --d;
       *bp = 'c';
       ap -= 8;
       bp -= 8;
       *ap = --d;
       *bp = 'b';
       ap -= 2;
       bp -= 2;
    }
  } /*else if*/

  /*5) keep track of c[k-1]-->c[k-3]-->...c[3]-->c[1]
  in <c[0],c[1],...,c[k-2],c[k-1]>*/
  i=k-1;
  s=0;
  while(i>1)
  {
    for(j=0;j<i-2;j++)
    {
      s += ca[j];
    }
    s += 1;
    if(c[i-2]>0)
    {
      d -= ca[i-1]+1;
      *ap = d;
      *bp = 'd';
      if(c[i]>0)
      {
        ap -= 4+5*ca[i-1];
        bp -= 4+5*ca[i-1];
      }
      else if(c[i]<0)
```

```
    {
      ap -= 7+5*ca[i-1];
      bp -= 7+5*ca[i-1];
    }
    *ap = --d;
    *bp = 'c';
    ap -= 6;
    bp -= 6;

    while(s<d)
    {
      *ap = --d;
      *bp = 'd';
      ap -= 4;
      bp -= 4;
      *ap = --d;
      *bp = 'c';
      ap -= 6;
      bp -= 6;
    }
  } /*if*/

  else if(c[i-2]<0)
  {
    d -= ca[i-1]+1;
    *ap = d;
    *bp = 'c';
    if(c[i]>0)
    {
      ap -= 5+5*ca[i-1];
      bp -= 5+5*ca[i-1];
    }
    else if(c[i]<0)
    {
      ap -= 8+5*ca[i-1];
      bp -= 8+5*ca[i-1];
    }
    *ap = --d;
    *bp = 'b';
    ap -= 2;
    bp -= 2;

    while(s<d)
    {
      *ap = --d;
      *bp = 'c';
```

```
        ap -= 8;
        bp -= 8;
        *ap = --d;
        *bp = 'b';
        ap -= 2;
        bp -= 2;
      }
    }/*else if*/
    i -= 2;
    s=0;
  } /*while*/

/*6) c[1]-->c[0] in <c[0],c[1],...,c[k-2],c[k-1]>*/
if(c[0]>0)
{
   d -= ca[0];
  *ap = d;
  *bp = 'b';
  if(c[1]>0)
  {
    ap += 1-5*ca[0];
    bp += 1-5*ca[0];
  }
  else if(c[1]<0)
  {
    ap -= 2+5*ca[0];
    bp -= 2+5*ca[0];
  }
}

else if(c[0]<0)
{
  d -= ca[0];
  *ap = d;
  *bp = 'c';
  if(c[1]>0)
  {
    ap += 2-5*ca[0];
    bp += 2-5*ca[0];
  }
  else if(c[1]<0)
  {
    ap -= 1+5*ca[0];
    bp -= 1+5*ca[0];
  }
}
```

```
/****reverse the orientation and track the codes***/

/*6') c[0]-->c[1] in <c[0],c[1],...,c[k-2],c[k-1]>*/
s = ca[0]+ca[1];
if(c[1]>0)
{
   d += ca[0];
  *ap = d;
  *bp = 'a';
  if(c[0]>0)
  {
    ap += 1+5*ca[0];
    bp += 1+5*ca[0];
  }
  else if(c[0]<0)
  {
    ap +=5*ca[0];
    bp +=5*ca[0];
  }
   *ap = ++d;
   *bp = 'b';
   ap += 6;
   bp += 6;
   while(s>d)
   {
      *ap = ++d;
      *bp = 'a';
      ap += 4;
      bp += 4;
      *ap =++d;
      *bp ='b';
      ap += 6;
      bp += 6;
   }
} /*if*/

else if(c[1]<0)
{
  d += ca[0];
  *ap = d;
  *bp = 'd';
  if(c[0]>0)
  {
    ap +=5*ca[0];
    bp +=5*ca[0];
```

```
  }
  else if(c[0]<0)
  {
    ap += -1+5*ca[0];
    bp += -1+5*ca[0];
  }
    *ap = ++d;
    *bp ='a';
    ap +=6;
    bp +=6;
    while(s>d)
    {
      *ap = ++d;
      *bp = 'd';
      ap += 4;
      bp += 4;
      *ap =++d;
      *bp ='a';
      ap += 6;
      bp += 6;
    }

}/*else if*/

/*5') c[1]->c[3]->...->c[k-1] in <c[0],c[1],...,c[k-2],c[k-1]>*/
i=1;
s=0;
while(i<k-2)
{
  for(j=0;j<i+3;j++)
  {
    s += ca[j];
  }
  if(c[i+2]>0)
  {
    d += ca[i+1]+1;
    *ap = d;
    *bp = 'a';
    if(c[i]>0)
    {
      ap += 4+5*ca[i+1];
      bp += 4+5*ca[i+1];
    }
    else if(c[i]<0)
    {
      ap += 5+5*ca[i+1];
```

```
    bp += 5+5*ca[i+1];
  }
  *ap = ++d;
 *bp = 'b';
 ap += 6;
 bp += 6;
 while(s>d)
 {
    *ap = ++d;
    *bp = 'a';
    ap += 4;
    bp += 4;
    *ap =++d;
    *bp ='b';
    ap += 6;
    bp += 6;
 }
} /*if*/

else if(c[i+2]<0)
{
  d += ca[i+1]+1;
  *ap = d;
  *bp = 'd';
  if(c[i]>0)
  {
    ap += 3+5*ca[i+1];
    bp += 3+5*ca[i+1];
  }
  else if(c[i]<0)
  {
    ap += 4+5*ca[i+1];
    bp += 4+5*ca[i+1];
  }
  *ap = ++d;
  *bp ='a';
  ap +=6;
  bp +=6;
  while(s>d)
  {
    *ap = ++d;
    *bp = 'd';
    ap += 4;
    bp += 4;
    *ap =++d;
    *bp ='a';
```

```
      ap += 6;
      bp += 6;
    }
  }/*else if*/
  i += 2;
  s=0;
} /*while*/

/*4') c[k-1]-->c[0] in <c[0],c[1],...,c[k-2],c[k-1]>*/
s=ca[0];
if(c[0]>0)
{
  d = 1;
  *ap = d;
  *bp = 'a';
  if(c[k-1]>0)
  {
    ap -= 1+5*(l-1);
    bp -= 1+5*(l-1);
  }
  else if(c[k-1]<0)
  {
    ap -= 5*(l-1);
    bp -= 5*(l-1);
  }
  *ap = ++d;
  *bp = 'd';
  ap += 4;
  bp += 4;
  while(s>d)
  {
     *ap = ++d;
     *bp = 'a';
     ap += 6;
     bp += 6;
     *ap = ++d;
     *bp = 'd';
     ap += 4;
     bp += 4;
  }
} /*if*/

else if(c[0]<0)
{
  d = 1;
  *ap = d;
```

```
  *bp = 'b';
  if(c[k-1]>0)
  {
    ap -= 5*(l-1);
    bp -= 5*(l-1);
  }
  else if(c[k-1]<0)
  {
    ap -= -1+5*(l-1);
    bp -= -1+5*(l-1);
  }
  *ap = ++d;
  *bp = 'a';
  ap += 4;
  bp += 4;
  while(s>d)
  {
     *ap = ++d;
     *bp = 'b';
     ap += 6;
     bp += 6;
     *ap = ++d;
     *bp = 'a';
     ap += 4;
     bp += 4;
  }
} /*else if*/

/*3') c[0]->c[2]->..->c[k-2] in <c[0],c[1],...,c[k-2],c[k-1]>*/
i = 0;
s=0;
while(i<k-2)
{
  for(j=0;j<i+3;j++)
  {
    s += ca[j];
  }
  if(c[i+2]>0)
  {
    d += ca[i+1]+1;
    *ap = d;
    *bp = 'a';
    if(c[i]>0)
    {
      ap += 1+5*(ca[i+1]+1);
      bp += 1+5*(ca[i+1]+1);
```

```
    }
    else if(c[i]<0)
    {
      ap += 5*(ca[i+1]+1);
      bp += 5*(ca[i+1]+1);
    }
    *ap = ++d;
    *bp = 'd';
    ap += 4;
    bp += 4;
  while(s>d)
  {
     *ap = ++d;
     *bp = 'a';
     ap += 6;
     bp += 6;
     *ap = ++d;
     *bp = 'd';
     ap += 4;
     bp += 4;
  }
} /*if*/

  else if(c[i+2]<0)
  {
    d += ca[i+1]+1;
    *ap = d;
    *bp = 'b';
    if(c[i]>0)
    {
      ap += 2+5*(ca[i+1]+1);
      bp += 2+5*(ca[i+1]+1);
    }
    else if(c[i]<0)
    {
      ap += 1+5*(ca[i+1]+1);
      bp += 1+5*(ca[i+1]+1);
    }
    *ap = ++d;
    *bp = 'a';
    ap += 4;
    bp += 4;
  while(s>d)
  {
     *ap = ++d;
     *bp = 'b';
```

```
      ap += 6;
      bp += 6;
      *ap = ++d;
      *bp = 'a';
      ap += 4;
      bp += 4;
   }

   }/*else if*/
   i += 2;
   s=0;
 } /*while*/

/*2') c[k-2]-->c[k-1] in <c[0],c[1],...,c[k-2],c[k-1]>*/
s = l-ca[k-1]+1;
if(c[k-1]>0)
{
  d += ca[k-1];
  *ap = d;
  *bp = 'a';
  if(c[k-2]>0)
  {
    ap += 1+5*ca[k-1];
    bp += 1+5*ca[k-1];
  }
  else if(c[k-2]<0)
  {
    ap += 5*ca[k-1];
    bp += 5*ca[k-1];
  }
  *ap = --d;
  *bp = 'b';
   ap -= 4;
   bp -= 4;

   while(s<d)
   {
     *ap = --d;
     *bp = 'a';
     ap -= 6;
     bp -= 6;
     *ap = --d;
     *bp = 'b';
     ap -= 4;
     bp -= 4;
   }
```

```
}  /*if*/

else if(c[k-1]<0)
{
  d += ca[k-1];
  *ap = d;
  *bp = 'd';
  if(c[k-2]>0)
  {
    ap += 5*ca[k-1];
    bp += 5*ca[k-1];
  }
  else if(c[k-2]<0)
  {
    ap += -1+5*ca[k-1];
    bp += -1+5*ca[k-1];
  }
  *ap = --d;
  *bp = 'a';
   ap -= 4;
   bp -= 4;

   while(s<d)
   {
     *ap = --d;
     *bp = 'd';
     ap -= 6;
     bp -= 6;
     *ap = --d;
     *bp = 'a';
     ap -= 4;
     bp -= 4;
   }
}

/*1') c[k-1]-->c[k-2]-->...-->c[1]
in <c[0],c[1],...,c[k-2],c[k-1]>*/
s=l-ca[k-1]+1;
for(i=k-1;i>0;i--)
{
  s -= ca[i-1];
  /*case 1*/
  if(c[i-1]>0 && (i-1)%2==1)
  {
       *ap = --d;
       *bp = 'a';
```

```
        if(c[i]>0)
        {
          ap -=4;
          bp -=4;
        }
        else if(c[i]<0)
        {
          ap -=5;
          bp -=5;
        }
        *ap = --d;
        *bp = 'b';
         ap -= 4;
         bp -= 4;

         while(s<d)
        {
          *ap = --d;
          *bp ='a';
           ap -=6;
           bp -=6;
          *ap = --d;
          *bp ='b';
           ap -=4;
           bp -=4;
        }
}

/*case 2*/
else if(c[i-1]<0 && (i-1)%2==1)
{
        *ap = --d;
        *bp = 'd';
        if(c[i]>0)
        {
          ap -=5;
          bp -=5;
        }
        else if(c[i]<0)
        {
          ap -=6;
          bp -=6;
        }
        *ap = --d;
        *bp = 'a';
         ap -= 4;
```

```
         bp -= 4;

        while(s<d)
        {
          *ap = --d;
        *bp ='d';
           ap -=6;
           bp -=6;
         *ap = --d;
        *bp ='a';
           ap -=4;
           bp -=4;
        }
   }

   /*case 3*/
   else if(c[i-1]>0 && (i-1)%2==0)
   {
        *ap = --d;
        *bp = 'a';
        if(c[i]>0)
        {
          ap -=6;
          bp -=6;
        }
        else if(c[i]<0)
        {
          ap -=5;
          bp -=5;
        }
        *ap = --d;
        *bp = 'd';
         ap -= 6;
         bp -= 6;
        while(s<d)
        {
          *ap = --d;
          *bp ='a';
           ap -=4;
           bp -=4;
          *ap = --d;
          *bp ='d';
           ap -=6;
           bp -=6;
        }
   }
```

```
  /*case 4*/
  else if(c[i-1]<0 && (i-1)%2==0)
  {
       *ap = --d;
       *bp = 'b';
       if(c[i]>0)
       {
         ap -=5;
         bp -=5;
       }
       else if(c[i]<0)
       {
         ap -=4;
         bp -=4;
       }
       *ap = --d;
       *bp = 'a';
        ap -= 6;
        bp -= 6;

        while(s<d)
       {
         *ap = --d;
         *bp ='b';
          ap -=4;
          bp -=4;
         *ap = --d;
         *bp ='a';
          ap -=6;
          bp -=6;
       }
  }
} /*for*/

/*EM code for unknot*/

if(k==2)
{
  if(c[0]==0 || c[1]==0)
  {
     int aa[]={1,2,2,1,1,2,1,1,2,2};
     char bb[]="+badc+badc";
     for(i=0;i<5*l+1;i++)
     {
       printf("aa[%d]=%d\n",i,aa[i]);
```

```
      }
      for(i=0;i<5*l+1;i++)
      {
        printf("bb[%d]=%c\n",i,bb[i]);
      }
     for(i=0;i<5*l+1;i++)
     {
      a[i]=aa[i];
      b[i]=bb[i];
     }
   }
}

for(i=0;i<l*5;i++)
{
   fprintf(f,"%d%c",a[i],b[i]);
  if(i%5==4)
   fprintf(f,"\n");
}
}/*end*/
```

B.2 Jones_2.m

```
%This program checks whether polynomials satisfy
%the seven properties of the Jones polynomial in section 4.2
%and signature conditions in lemma 4.3 and lemma 4.4 or not.

%This is used to determine whether vkp1 and vkp2 in theorem 4.1
%and theorem 4.2 could be the Jones polynomial of knots.
%The polynomials vkp1 and vkp2 are calculated using
%the Jones polynomial vk1 and vk2 of two knots k1 and k2
%whose knot distance has lower bound 1.

clear all;

poly = importdata('JonesPoly13_new.txt',';',1552);
sig = importdata('Signature.txt');
p = importdata('pair.txt');

syms t;
[row col] = size(p);

fid= fopen('result.txt', 'w');
NewSign = fopen('NewSign.txt', 'w');
```

```
Change = fopen('Changed.txt', 'w');
fvkp1 = fopen('vkp1.txt','w');
fvkp2 = fopen('vkp2.txt','w');
fw = fopen('wk.txt','w');
f1 = fopen('con7.txt','w');
f7rc = fopen('con7rc.txt', 'w');
f3 = fopen('con3.txt', 'w');

for n =1:row
    k1=0;   %track whether vkp1 violates one of the properties
    k2=0;   %track whether vkp2 violates one of the properties
    vk1 = sym(poly(p(n,1)));
    vk2 = sym(poly(p(n,2)));

   %vkp1 = vk' in theorem 4.1
    vt1 = vk2 - (factor((vk1-vk2)/(t-1)));
    vt1d_1 = subs(diff(vt1), t, 1);
    vkp1 = expand(t^-vt1d_1 * vt1);
    vkp1_str=char(vkp1);
    fprintf(fvk1,'%d ,%d\n', p(n,1)-1, p(n,2)-1);
    fprintf(fvk1,'%s\n',vkp1_str);
    vkp1_1d=subs(diff(vkp1),t,1);
    vkp1_2d=subs(diff(vkp1,2),t,1);
    vkp1_3d=subs(diff(vkp1,3),t,1);

    %vkp2 = vk' in theorem 4.2
    vt2 = vk1 - (factor((vk2-vk1)/(t-1)));
    vt2d_1 = subs(diff(vt2), t, 1);
    vkp2 = expand(t^-vt2d_1 * vt2);
    vkp2_str=char(vkp2);
    fprintf(fvk2,'%d ,%d\n', p(n,1)-1, p(n,2)-1);
    fprintf(fvk2,'%s\n',vkp2_str);
    vkp2_1d=subs(diff(vkp2),t,1);
    vkp2_2d=subs(diff(vkp2,2),t,1);
    vkp2_3d=subs(diff(vkp2,3),t,1);

    %check property (1) in section 4.2
    if(subs(vkp1, t, 1) ≠ 1 && k1==0)
     fprintf(fid, '(1)-1 %d, %d (%d) \n', p(n,1)-1,p(n,2)-1,n);
     k1=k1+1;
     if(k1==1 && k2==1)
     fprintf(fid,'\n');
     fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)-1,p(n,2)-1);
     fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)-1,p(n,2)-1);
     fprintf(NewSign,'else\n');
     fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
```

```
 contradicts to existing value)\n\n', p(n,1)−1, p(n,2)−1);
 fprintf(Change,'Changed[%d][%d]=1;\n', p(n,1)−1,p(n,2)−1);
 continue;
 end
end

if(subs(vkp2, t, 1) ≠ 1 && k2==0)
 fprintf(fid, '(1)−2 %d, %d (%d) \n', p(n,1)−1, p(n,2)−1,n);
 k2=k2+1;
 if(k1==1 && k2==1)
 fprintf(fid,'\n');
 fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)−1,p(n,2)−1);
 fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)−1,p(n,2)−1);
 fprintf(NewSign,'else\n');
 fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
 contradicts to existing value)\n\n', p(n,1)−1, p(n,2)−1);
 fprintf(Change,'Changed[%d][%d]=1;\n', p(n,1)−1,p(n,2)−1);
 continue;
 end
end


%check property (2) in section 4.2
if(vkp1_1d ≠ 0 && k1==0)
   fprintf(fid, '(2)−1 %d, %d (%d)\n', p(n,1)−1,p(n,2)−1,n);
   k1=k1+1;
  if(k1==1 && k2==1)
   fprintf(fid,'\n');
   fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)−1,p(n,2)−1);
   fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)−1,p(n,2)−1);
   fprintf(NewSign,'else\n');
   fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
   contradicts to existing value)\n\n', p(n,1)−1, p(n,2)−1);
   fprintf(Change,'Changed[%d][%d]=1;\n', p(n,1)−1,p(n,2)−1);
   continue;
  end
end

if(vkp2_1d ≠ 0 && k2==0)
  fprintf(fid, '(2)−2 %d, %d (%d)\n', p(n,1)−1,p(n,2)−1,n);
  k2=k2+1;
 if(k1==1 && k2==1)
  fprintf(fid,'\n');
  fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)−1,p(n,2)−1);
  fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)−1,p(n,2)−1);
  fprintf(NewSign,'else\n');
```

```
   fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
   contradicts to existing value)\n\n', p(n,1)-1, p(n,2)-1);
   fprintf(Change,'Changed[%d][%d]=1;\n', p(n,1)-1,p(n,2)-1);
   continue;
  end
 end

 %check property (3) in section 4.2
   p3_1= subs(vkp1, t, exp(2*pi*1i/3));
   p3_2= subs(vkp2, t, exp(2*pi*1i/3));
if(k1==0)
if(round(imag(p3_1)) ≠0 || round(real(p3_1)) ≠1)
  fprintf(fid, '(3)-1 %d,%d (%d) \n',p(n,1)-1,p(n,2)-1,n);
  fprintf(f3, '(3)-1 %d,%d (%d) == %g+%gi \n',p(n,1)-1,
  p(n,2)-1,n,real(p3_1),imag(p3_1));
  k1=k1+1;
 if(k1==1 && k2==1)
  fprintf(fid,'\n');
  fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)-1,p(n,2)-1);
  fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)-1,p(n,2)-1);
  fprintf(NewSign,'else\n');
  fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
  contradicts to existing value)\n\n', p(n,1)-1, p(n,2)-1);
  fprintf(Change,'Changed[%d][%d]=1;\n', p(n,1)-1,p(n,2)-1);
  continue;
 end
end
end

if(k2==0)
if(round(imag(p3_2)) ≠0  || round(real(p3_2))≠1)
  fprintf(fid, '(3)-2 %d,%d (%d) \n',p(n,1)-1,p(n,2)-1,n);
  fprintf(f3, '(3)-2 %d,%d (%d) == %g+%gi \n',p(n,1)-1,
  p(n,2)-1,n,real(p3_2),imag(p3_2));
  k2=k2+1;
 if(k1==1 && k2==1)
  fprintf(fid,'\n');
  fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)-1,p(n,2)-1);
  fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)-1,p(n,2)-1);
  fprintf(NewSign,'else\n');
  fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
  contradicts to existing value)\n\n', p(n,1)-1, p(n,2)-1);
  fprintf(Change,'Changed[%d][%d]=1;\n', p(n,1)-1,p(n,2)-1);
  continue;
 end
end
```

```
end

 %check property (4) in section 4.2
 if(k1==0)
  if(rem(vkp1_3d,18) ≠0)
    fprintf(fid, '(4)−1 %d, %d (%d)\n', p(n,1)−1,p(n,2)−1,n);
    k1=k1+1;
   if(k1==1 && k2==1)
    fprintf(fid,'\n');
    fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)−1,p(n,2)−1);
    fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)−1,p(n,2)−1);
    fprintf(NewSign,'else\n');
    fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
    contradicts to existing value)\n\n', p(n,1)−1, p(n,2)−1);
    fprintf(Change,'Changed[%d][%d]=1;\n', p(n,1)−1,p(n,2)−1);
    continue;
   end
  end
 end

 if(k2==0)
  if(rem(vkp2_3d,18) ≠0)
    fprintf(fid, '(4)−2 %d, %d (%d)\n', p(n,1)−1,p(n,2)−1,n);
    k2=k2+1;
   if(k1==1 && k2==1)
    fprintf(fid,'\n');
    fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)−1,p(n,2)−1);
    fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)−1,p(n,2)−1);
    fprintf(NewSign,'else\n');
    fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
    contradicts to existing value)\n\n', p(n,1)−1, p(n,2)−1);
    fprintf(Change,'Changed[%d][%d]=1;\n', p(n,1)−1,p(n,2)−1);
    continue;
   end
  end
 end

 %check property (5) in section 4.2
if(k1==0)
 if(rem(vkp1_3d+3*vkp1_2d,36) ≠0 )
   fprintf(fid, '(5)−1 %d, %d (%d)\n', p(n,1)−1,p(n,2)−1,n);
   k1=k1+1;
  if(k1==1 && k2==1)
   fprintf(fid,'\n');
   fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)−1,p(n,2)−1);
   fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)−1,p(n,2)−1);
```

```
    fprintf(NewSign,'else\n');
    fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
    contradicts to existing value)\n\n', p(n,1)-1, p(n,2)-1);
    fprintf(Change,'Changed[%d][%d]=1;\n', p(n,1)-1,p(n,2)-1);
    continue;
   end
 end
 end

 if(k2==0)
 if(rem(vkp2_3d+3*vkp2_2d,36) ≠0)
   fprintf(fid, '(5)-2 %d, %d (%d)\n', p(n,1)-1,p(n,2)-1,n);
   k2=k2+1;
  if(k1==1 && k2==1)
  fprintf(fid,'\n');
  fprintf(NewSign,'if(Sign[%d][%d]==1)\n',p(n,1)-1,p(n,2)-1);
  fprintf(NewSign,'Sign[%d][%d]=2;\n', p(n,1)-1,p(n,2)-1);
  fprintf(NewSign,'else\n');
  fprintf(NewSign,'fprintf(mistake,Sign[%d][%d]=2
  contradicts to existing value)\n\n', p(n,1)-1, p(n,2)-1);
  fprintf(Change, 'Changed[%d][%d]=1;\n', p(n,1)-1,p(n,2)-1);
  continue;
 end
 end
 end

  %check property (6) in section 4.2
  f6=expand(factor((1-vkp1)/((1-t)*(1-t^3))));
  f6_str=char(f6);

  f7=expand(factor((1-vkp2)/((1-t)*(1-t^3))));
  f7_str=char(f7);

  fprintf(fw,'%d ,%d (%d)\n', p(n,1)-1, p(n,2)-1,n);
  fprintf(fw,'%s\n',f6_str);
  fprintf(fw,'%s\n',f7_str);

 %check property (7) in section 4.2
  c7_1=subs(vkp1, exp(pi*1i/3));
  c7_2=subs(vkp2, exp(pi*1i/3));
 if(k1==0)
  if(round(real(c7_1)) ≠0 && round(imag(c7_1)) ≠0)
   fprintf(fid, '(7)-1 %d, %d (%d)\n', p(n,1)-1,p(n,2)-1,n);
   fprintf(f1, '(7)-1 (%d,%d)=%g+%gi\n',p(n,1)-1,
         p(n,2)-1, real(c7_1),imag(c7_1));
   k1=k1+1;
```

```
 if(k1==1 && k2==1)
  fprintf(fid,'\n');
  fprintf(NewSign, 'if(Sign[%d][%d]==1)\n',p(n,1)-1,p(n,2)-1);
  fprintf(NewSign, 'Sign[%d][%d]=2;\n', p(n,1)-1,p(n,2)-1);
  fprintf(NewSign, 'else\n');
  fprintf(NewSign, 'fprintf(mistake,Sign[%d][%d]=2
  contradicts to existing value)\n\n', p(n,1)-1, p(n,2)-1);
  fprintf(Change, 'Changed[%d][%d]=1;\n', p(n,1)-1,p(n,2)-1);
  continue;
  end
  else
       fprintf(f7rc,'(7)-1 (%d,%d)=%g+%gi\n',p(n,1)-1,
       p(n,2)-1, real(c7_1),imag(c7_1));
  end
  end

  if(k2==0)
  if(round(real(c7_2)) ≠0 && round(imag(c7_2)) ≠0)
    fprintf(fid, '(7)-2 %d, %d (%d)\n', p(n,1)-1,p(n,2)-1,n);
    fprintf(f1,'(7)-2 (%d,%d)=%g+%gi\n',p(n,1)-1,
        p(n,2)-1, real(c7_2),imag(c7_2));
    k2=k2+1;
   if(k1==1 && k2==1)
    fprintf(fid,'\n');
    fprintf(NewSign, 'if(Sign[%d][%d]==1)\n',p(n,1)-1,p(n,2)-1);
    fprintf(NewSign, 'Sign[%d][%d]=2;\n', p(n,1)-1,p(n,2)-1);
    fprintf(NewSign, 'else\n');
    fprintf(NewSign, 'fprintf(mistake,Sign[%d][%d]=2 contradicts
    to existing value)\n\n', p(n,1)-1, p(n,2)-1);
    fprintf(Change, 'Changed[%d][%d]=1;\n', p(n,1)-1,p(n,2)-1);
    continue;
   end
  else
       fprintf(f7rc,'(7)-2 (%d,%d)=%g+%gi\n',p(n,1)-1,
       p(n,2)-1, real(c7_1),imag(c7_1));
  end
  end

  % check signature
  s1 = sig(p(n,1));
  s2 = sig(p(n,2));
  s3= fopen('NewbySignature.txt', 'w');
if(s1 ≠ -100 && s2 ≠ -100)
 if(s2-s1>2)
  fprintf(s3,'(1) %d,%d \n', p(n,1)-1, p(n,2)-1);
   if(k1==0)
```

```
      fprintf(fid, '(s)-1 %d, %d (%d)\n', p(n,1)-1,p(n,2)-1,n);
      k1=k1+1;
     if(k1==1 && k2==1)
      fprintf(fid,'\n');
      fprintf(NewSign, 'if(Sign[%d][%d]==1)\n',p(n,1)-1,p(n,2)-1);
      fprintf(NewSign, 'Sign[%d][%d]=2;\n', p(n,1)-1,p(n,2)-1);
      fprintf(NewSign, 'else\n');
      fprintf(NewSign, 'fprintf(mistake,Sign[%d][%d]=2 contradicts
      to existing value)\n\n', p(n,1)-1, p(n,2)-1);
      fprintf(Change, 'Changed[%d][%d]=1;\n', p(n,1)-1,p(n,2)-1);
      continue;
      end
      end
     end
   if(s2-s1>2)
      fprintf(s3,'(2) %d,%d \n', p(n,1)-1, p(n,2)-1);
    if(k2==0)
     fprintf(fid, '(s)-2 %d, %d (%d)\n', p(n,1)-1,p(n,2)-1,n);
     k2=k2+1;
    if(k1==1 && k2==1)
     fprintf(fid,'\n');
     fprintf(NewSign, 'if(Sign[%d][%d]==1)\n',p(n,1)-1,p(n,2)-1);
     fprintf(NewSign, 'Sign[%d][%d]=2;\n', p(n,1)-1,p(n,2)-1);
     fprintf(NewSign, 'else\n');
     fprintf(NewSign, 'fprintf(mistake,Sign[%d][%d]=2
     contradicts to existing value)\n\n', p(n,1)-1, p(n,2)-1);
     fprintf(Change, 'Changed[%d][%d]=1;\n', p(n,1)-1,p(n,2)-1);
     continue;
    end
    end
    end
    end
    fprintf(fid,'\n');
end  %for%

fclose(fid);
fclose(fvkp1);
fclose(fvkp2);
fclose(fw);
fclose(NewSign);
fclose(Change);
fclose(f1);
fclose(f7rc);
fclose(f3);
```

APPENDIX C
NEW KNOT DISTANCE TABLE

Table C.1: New knot distance table.

New Knot Distance Table.

	0_1	3_1	4_1	5_1	5_2	6_1	6_2	6_3	$3_1 \# 3_1$	$3_1 \# 3_1^*$	7_1	7_2	7_3	7_4
3_1	1	0	2	1	1	2	1	1	1	1	2	2	3	2-3
3_1^*	1	2	2	3	2	2	2	1	3	1	4	2	2	1
4_1	1	2	0	2-3	2	1	1	2	2-3	2-3	3-4	2	2-3	2-3
5_1	2	1	2-3	0	1	2-3	2	2	2	2	1	2	4	3-4
5_1^*	2	3	2-3	4	3	2-3	3	2	4	2	5	3	1	2
5_2	1	1	2	1	0	2	2	2	2	2	2	1	3	2-3
5_2^*	1	2	2	3	2	2	2	2	3	2	4	2	1	1
6_1	1	2	1	2-3	2	0	1	2	2-3	1-3	3-4	2	2-3	2-3
6_1^*	1	2	1	2-3	2	1	2	2	2-3	1-3	3-4	2	2-3	2-3
6_2	1	1	1	2	2	1	0	2	2	2	2-3	2	3	2-3
6_2^*	1	2	1	3	2	2	2	2	3	2	4	2	2-3	2
6_3	1	1	2	2	2	2	2	0	2	2	3	2	2-3	2
$3_1 \# 3_1$	2	1	2-3	2	2	2-3	2	2	0	2	2-3	2-3	4	3-4
$3_1^* \# 3_1^*$	2	3	2-3	4	3	2-3	3	2	4	2	5	3	2-3	2
$3_1 \# 3_1^*$	2	1	2-3	2	2	1-3	2	2	2	0	3	2-3	2-3	2
7_1	3	2	3-4	1	2	3-4	2-3	3	2-3	3	0	2	5	4-5
7_1^*	3	4	3-4	5	4	3-4	4	3	5	3	6	4	1	2
7_2	1	2	2	2	1	2	2	2	2-3	2-3	2	0	3	2-3
7_2^*	1	2	2	3	2	2	2	2	3	2-3	4	2	1	2
7_3	2	3	2-3	4	3	2-3	3	2-3	4	2-3	5	3	0	1
7_3^*	2	2	2-3	1	1	2-3	2-3	2-3	2-3	2-3	1	1	4	3-4
7_4	2	2-3	2-3	3-4	2-3	2-3	2-3	2	3-4	2	4-5	2-3	1	0
7_4^*	2	1	2-3	2	1	2-3	2	2	2	2	2	2	3-4	2-4
7_5	2	1	2-3	1	1	2-3	2	2	2	2	1	1	4	3-4
7_5^*	2	3	2-3	4	3	2-3	3	2	4	2	5	3	2	2
7_6	1	1	1	2	1	2	2	2	2	2	2-3	1	3	2-3
7_6^*	1	2	1	3	2	2	2	2	3	2	4	2	2	2
7_7	1	2	1	2-3	2	2	2	2	2-3	1-2	3-4	2	2-3	2
7_7^*	1	1	1	2	2	2	2	2	2	1-2	3	2	2-3	2-3
$3_1 \# 4_1$	2	1	1	1-2	1-2	2	2	2	2	2	2-3	2-3	3-4	2-4
$3_1^* \# 4_1$	2	2-3	1	3-4	2-3	2	2	2	3-4	2	4-5	2-3	1-3	1-2
8_1	1	2	2	2-3	2	1	2	2	2-3	2-3	3-4	2	2-3	2-3
8_1^*	1	2	2	2-3	2	2	2	2	2-3	2-3	3-4	2	2-3	2-3
8_2	2	1	2	1	2	2	1	2	2	2	2	2-3	4	3-4
8_2^*	2	3	2	4	3	2-3	3	2	4	2	5	3	2	2
8_3	2	2-3	2	2-4	2-3	1	2	2-3	2-4	2-4	3-5	2-3	2-4	2-4
8_4	2	2	1	2-3	2-3	2	1	2-3	2-3	2-3	2-4	2-3	3-4	2-4
8_4^*	2	2-3	1	3-4	2-3	1	2	2-3	3-4	2-3	4-5	2-3	2-4	2-3
8_5	2	3	2	4	3	2-3	3	2-3	4	2-3	5	3	1-4	2-3
8_5^*	2	1-2	2	2-3	1-3	2	1	2-3	1	2-3	1-4	1-3	4	3-4
8_6	2	1	2	2	2	1	1	2	2	2	2-3	2-3	3-4	2-4
8_6^*	2	2-3	2	3-4	2-3	2	2-3	2	3-4	2	4-5	2-3	2-3	2
8_7	1	2	2	3	2	2	2	1	3	2	4	2	2	2
8_7^*	1	1	2	1	2	2	2	1	2	2	2	2	3	2-3
8_8	2	1	2-3	2	2	2-3	2	1	2	2	3	2-3	2	2
8_8^*	2	1	2-3	2	1	2-3	2	1	2	2	3	2	2-3	2
8_9	1	2	2	2-3	2	2	1	2	2-3	2-3	3-4	2	2-3	2-3

Table C.1: Continued.

New Knot Distance Table (cont.)

	7_5	7_6	7_7	3_1 # 4_1	8_1	8_2	8_3	8_4	8_5	8_6	8_7	8_8	8_9	8_{10}	8_{11}
3_1	1	1	2	1	2	1	2-3	2	3	1	2	1	2	2	2
3_1^*	3	2	1	2-3	2	3	2-3	2-3	1-2	2-3	1	1	2	1-2	2
4_1	2-3	1	1	1	2	2	2	1	2	2	2	2-3	2	2-3	2
5_1	1	2	2-3	1-2	2-3	1	2-4	2-3	4	2	3	2	2-3	3	2
5_1^*	4	3	2	3-4	2-3	4	2-4	3-4	2-3	3-4	1	2	2-3	1	3
5_2	1	1	2	1-2	2	2	2-3	2-3	3	2	2	2	2	2-3	1
5_2^*	3	2	2	2-3	2	3	2-3	2-3	1-3	2-3	2	1	2	1	2
6_1	2-3	2	2	2	1	2	1	2	2-3	1	2	2-3	2	1-3	1
6_1^*	2-3	2	2	2	2	2-3	1	1	2	2	2	2-3	2	1-3	2
6_2	2	2	2	2	2	1	2	1	3	1	2	2	1	2-3	1
6_2^*	3	2	2	2	2	3	2	2	1	2-3	2	2	1	1-3	2
6_3	2	2	2	2	2	2	2-3	2-3	2-3	2	1	1	2	1	2
3_1 # 3_1	2	2	2-3	2	2-3	2	2-4	2-3	4	2	3	2	2-3	3	1-3
3_1^* # 3_1^*	4	3	2	3-4	2-3	4	2-4	3-4	1	3-4	2	2	2-3	1-3	3
3_1 # 3_1^*	2	2	1-2	2	2-3	2	2-4	2-3	2-3	2	2	2	2-3	1	1-3
7_1	1	2-3	3-4	2-3	3-4	2	3-5	2-4	5	2-3	4	3	3-4	4	2-3
7_1^*	5	4	3	4-5	3-4	5	3-5	4-5	1-4	4-5	2	3	3-4	2	4
7_2	1	1	2	2-3	2	2-3	2-3	2-3	3	2-3	2	2-3	2	2-3	2
7_2^*	3	2	2	2-3	2	3	2-3	2-3	1-3	2-3	2	2	2	1-2	2
7_3	4	3	2-3	3-4	2-3	4	2-4	3-4	1-4	3-4	2	2	2-3	2	3
7_3^*	2	2	2-3	1-3	2-3	2	2-4	2-4	4	2-3	3	2-3	2-3	3-4	2
7_4	3-4	2-3	2	2-4	2-3	3-4	2-4	2-4	2-3	2-4	2	2	2-3	1-2	2-3
7_4^*	2	2	2-3	1-2	2-3	2	2-4	2-3	3-4	2	2-3	2	2-3	2-3	2
7_5	0	2	2-3	2	2-3	2	2-4	2-3	4	2	3	2	2-3	3	2
7_5^*	4	3	2	3-4	2-3	4	2-4	3-4	2-3	3-4	2	2	2-3	1-2	3
7_6	2	0	2	2	2	2	2-3	2	3	2	2	2	2	2-3	2
7_6^*	3	2	2	2	2	3	2-3	2	1-3	2-3	2	2	2	1-2	2
7_7	2-3	2	0	2	2	2-3	2-3	2	2-3	2-3	2	2	2	1-3	2
7_7^*	2	2	2	2	2	2	2-3	2	2-3	2	2	2	2	1-3	2
3_1 # 4_1	2	2	2	0	2-3	2	1-3	2	3	1-2	2-3	2	2-3	2-3	2-3
3_1^* # 4_1	3-4	2	2	2	2-3	3	1-3	2	2-3	2-3	1-2	2	2-3	1-3	2-3
8_1	2-3	2	2	2-3	0	2	1	2	2-3	1	2	2-3	2	1-3	1
8_1^*	2-3	2	2	2-3	2	2-3	1	2	2-3	2-3	2	2-3	2	1-3	2
8_2	2	2	2-3	2	2	0	2-3	2	4	1	3	2	2	3	1
8_2^*	4	3	2	3	2-3	4	2-3	3	2	3-4	2	2	2	1-2	3
8_3	2-4	2-3	2-3	1-3	1	2-3	0	1	2-3	2	2-3	2-4	2	2-4	2
8_4	2-3	2	2	2	2	2	1	0	3	1	2-3	2-3	1	2-4	2
8_4^*	3-4	2	2	2	2	3	1	2	1-2	2	2-3	2-3	1	1-4	2
8_5	4	3	2-3	3	2-3	4	2-3	3	0	3-4	1-3	2-3	2	1-4	3
8_5^*	2-3	1-3	2-3	2-3	2-3	2	2-3	1-2	4	1-2	3	2-3	2	3-4	2
8_6	2	2	2-3	1-2	1	1	2	1	3-4	0	2-3	2	2	2-3	2
8_6^*	3-4	2-3	2	2-3	2-3	3-4	2	2	1-2	2-3	2	2	2	1-3	2-3
8_7	3	2	2	2-3	2	3	2-3	2-3	1-3	2-3	0	1	2	1-2	2
8_7^*	2	2	2	1-2	2	2	2-3	2-3	3	2	2	2	2	2	2
8_8	2	2	2	2	2-3	2	2-4	2-3	2-3	2	1	0	2-3	1-2	2-3
8_8^*	2	2	2	2	2-3	2	2-4	2-3	2-3	2	2	1	2-3	1-2	2
8_9	2-3	2	2	2-3	2	2	2	1	2	2	2	2-3	0	2-3	2

Table C.1: Continued.

New Knot Distance Table (cont.)

	0_1	3_1	4_1	5_1	5_2	6_1	6_2	6_3	$3_1 \# 3_1$	$3_1 \# 3_1^*$	7_1	7_2	7_3	7_4
8_{10}	1-2	2	2-3	3	2-3	1-3	2-3	1	3	1	4	2-3	2	1-2
8_{10}^*	1-2	1-2	2-3	1	1	1-3	1-3	1	1-3	1	2	1-2	3-4	2-3
8_{11}	1	2	2	2	1	1	1	2	1-3	1-3	2-3	2	3	2-3
8_{11}^*	1	2	2	3	2	2	2	2	3	1-3	4	2	2	2
8_{12}	2	2-3	1	2-4	2-3	1	2	2-3	2-4	2-4	3-5	2-3	2-4	2-4
8_{13}	1	2	1	2	1	2	2	1	2-3	2-3	3	2	2-3	2-3
8_{13}^*	1	2	1	2-3	2	2	2	1	2-3	2-3	3-4	2	2	2
8_{14}	1	1	1	2	1	2	1	2	2	2	2-3	2	3	2-3
8_{14}^*	1	2	1	3	2	2	2	2	3	2	4	2	2	2
8_{15}	2	1-2	2-3	1	1	2-3	1-3	2-3	1	2-3	1-2	1-2	4	3-4
8_{15}^*	2	3	2-3	4	3	2-3	3	2-3	4	2-3	5	3	1-2	1-2
8_{16}	1-2	1	1	1-2	1	1-2	1-2	1	2	2	2-3	1-2	3-4	2-3
8_{16}^*	1-2	2	1	3	2-3	1-2	2	1	3	2	4	2-3	2	1-2
8_{17}	1	1	2	2	1-2	1-2	1-2	2	2	2	3	1-2	2-3	1-2
8_{18}	2	1	2-3	2	2	1-3	2	2	2	1-2	3	2-3	2-3	1-2
8_{19}	3	4	3-4	5	4	3-4	4	3	5	3	6	4	2	2-3
8_{19}^*	3	2	3-4	1	2	3-4	2-3	3	1	3	1-2	2-3	5	4-5
8_{20}	1	1-2	1-2	2	1	1-2	1-2	1	2-3	1	3	1-2	2-3	2-3
8_{20}^*	1	1-2	1-2	2-3	1-2	1-2	1-2	1	2-3	1	3-4	1-2	2	2
8_{21}	1	1-2	1-2	1-3	1-2	2	1	2	1	1-3	2-4	1-2	3	2-3
8_{21}^*	1	2	1-2	3	2	2	2	2	3	1-3	4	2	1-3	1-3
$3_1 \# 5_1$	3	2	3-4	1	2	3-4	2-3	3	1	3	2	2-3	5	4-5
$3_1^* \# 5_1^*$	3	4	3-4	5	4	3-4	4	3	5	3	6	4	2	2-3
$3_1 \# 5_1^*$	1-3	2	1-4	3	2-3	2-4	2-3	2-3	3	1	4	2-4	2	1-3
$3_1^* \# 5_1$	1-3	2	1-4	1	2	2-4	1-3	2-3	1-3	1-3	2	1-3	3-4	2-3
$3_1 \# 5_2$	2	1	2-3	2	1	2-3	1-2	2	1	2	1-3	1-2	4	3-4
$3_1^* \# 5_2^*$	2	3	2-3	4	3	2-3	3	2	4	2	5	3	1-2	2
$3_1 \# 5_2^*$	1-2	1	1-3	2	1-2	2-3	2	2	2	1	3	2-3	2	2
$3_1^* \# 5_2$	1-2	1-2	1-3	2	1	2-3	2-3	2	2-3	1-2	3	2	2-3	2
$4_1 \# 4_1$	2	2-3	1	2-4	2-3	2	2	2-3	2-4	2-4	3-5	2-3	2-4	1-4
9_1	4	3	4-5	2	3	4-5	3-4	4	2-4	4	2-1	3	6	5-6
9_1^*	4	5	4-5	6	5	4-5	5	4	6	4	7	5	2	3
9_2	1	2	2	2-3	2	2	2	2	1-3	1-3	2-3	1	3	2-3
9_2^*	1	2	2	3	2	2	2	2	3	1-3	4	2	2	2-3
9_3	3	4	3-4	5	4	3-4	4	3-4	5	3-4	6	4	1	2
9_3^*	3	2-3	3-4	2	2	3-4	2-4	3-4	2-4	3-4	1	2	5	4-5
9_4	2	2-3	2-3	2	2	2-3	2-3	2-3	1-4	2-4	2	1	4	3-4
9_4^*	2	3	2-3	4	3	2-3	3	2-3	4	2-4	5	3	1	2
9_5	2	2-3	2-3	3-4	2-3	2-3	2-3	2-3	3-4	2-3	4-5	2-3	2	1
9_5^*	2	2	2-3	2	1	2-3	2-3	2-3	2-3	2-3	2-3	1	3-4	2-4
9_6	3	2	3-4	1	2	3-4	2-3	3	1-3	3	1	2	5	4-5
9_6^*	3	4	3-4	5	4	3-4	4	3	5	3	6	4	2	2-3
9_7	2	1	2-3	2	2	2-3	2	2	2	2	2	1	4	3-4
9_7^*	2	3	2-3	4	3	2-3	3	2	4	2	5	3	2	2
9_8	2	1	1	2	2	2	2	2	2	2	2-3	2	3-4	2-4
9_8^*	2	2-3	1	3-4	2-3	1	2	2	3-4	2	4-5	2-3	2-3	2
9_9	3	2	3-4	2	2	3-4	2-3	3	2-3	3	1	2	5	4-5

Table C.1: Continued.

New Knot Distance Table (cont.)

	7_5	7_6	7_7	3_1 # 4_1	8_1	8_2	8_3	8_4	8_5	8_6	8_7	8_8	8_9	8_{10}	8_{11}
8_{10}	3	2-3	1-3	2-3	1-3	3	2-4	2-4	1-4	2-3	1-2	1-2	2-3	0	2-3
8_{10}^*	1-2	1-2	1-3	1-3	1-3	1-2	2-4	1-4	3-4	1-3	2	1-2	2-3	2	1-2
8_{11}	2	2	2	2-3	1	1	2	2	3	2	2	2-3	2	2-3	0
8_{11}^*	3	2	2	2-3	2	3	2	2	2	2-3	2	2	2	1-2	2
8_{12}	2-4	2	2	2	1	2-3	2	2	2-3	2	2-3	2-4	2-3	2-4	2
8_{13}	2	2	2	2	2	2-3	2-3	2	2-3	2-3	2	2	2	2	2
8_{13}^*	2-3	2	2	2	2	2-3	2-3	2	2-3	2-3	1	2	2	2	2
8_{14}	2	2	2	2	2	1	2-3	2	3	2	2	2	2	2-3	2
8_{14}^*	3	2	2	2	2	3	2-3	2	1-2	2-3	2	2	2	1-2	2
8_{15}	2	1-2	2-3	1-3	2-3	2	2-4	1-4	4	1-3	3	2-3	2-3	3-4	2
8_{15}^*	4	3	2-3	3-4	2-3	4	2-4	3-4	1-2	3-4	1-2	2	2-3	1-2	3
8_{16}	2	1-2	1-2	1-2	1-3	2	1-3	1-2	3	1-2	2	2	2-3	2	1-2
8_{16}^*	3	2	1-2	2	1-3	3	1-3	2	1-3	2-3	1-2	2	2-3	1-2	2-3
8_{17}	2	1-2	1-2	1-2	2	2	2-3	1-3	2	1-2	1-2	1-2	2	1-3	1-2
8_{18}	2	2	1-2	1-2	2-3	2	2-4	2-3	2-3	2	2	2	2-3	1-3	1-3
8_{19}	5	4	3	4-5	3-4	5	3-5	4-5	1-2	4-5	2	3	3-4	2	4
8_{19}^*	1-2	2-3	3-4	2-3	3-4	1-2	3-5	2-4	5	2-3	4	3	3-4	4	2-3
8_{20}	2	1-2	1-2	2-3	1-2	2-3	1-3	1-3	2-3	1-3	1-2	2	1-2	1-2	1-2
8_{20}^*	2-3	1-2	1-2	2-3	1-2	2-3	1-3	1-3	2-3	1-3	1-2	2	1-2	1-2	1-2
8_{21}	2-3	1-2	1-2	1-3	2	2	1-3	1-2	3	1-2	2	2-3	2	2-3	2
8_{21}^*	3	2	1-2	2-3	2	3	1-3	2-3	1-2	2-3	1-2	2-3	2	1-3	2
3_1 # 5_1	2	2-3	3-4	2-3	3-4	2	3-5	2-4	5	2-3	4	3	3-4	4	2-3
3_1^* # 5_1^*	5	4	3	4-5	3-4	5	3-5	4-5	1-2	4-5	2	3	3-4	2	4
3_1 # 5_1^*	3	2-3	1-3	2-3	2-4	3	2-5	2-4	1-4	2-3	2	2-3	2-4	2	2-4
3_1^* # 5_1	2	1-3	1-3	1-3	2-4	2	2-5	1-4	3-4	2-3	2-3	2-3	2-4	2-4	2-3
3_1 # 5_2	2	1-2	2-3	2	2-3	2	2-4	1-3	4	1-2	3	2	2-3	3	2
3_1^* # 5_2^*	4	3	2	3-4	2-3	4	2-4	3-4	1-2	3-4	1-2	2	2-3	1-2	3
3_1 # 5_2^*	2	2	1-3	2	2-3	2	1-4	2-3	2-4	2	2-3	2	1-3	1-2	2-3
3_1^* # 5_2	2	2	1-2	2-3	2-3	2-3	1-4	2-4	2-3	2-3	2	2	1-3	1-3	2
4_1 # 4_1	2-4	2	2	1-2	2-3	2-3	2-3	2	2-3	2-3	2-3	1-4	1-3	2-4	2-3
9_1	2	3-4	4-5	3-4	4-5	2-3	4-6	3-5	6	3-4	5	4	4-5	5	3-4
9_1^*	6	5	4	5-6	4-5	6	4-6	5-6	2-5	5-6	3	4	4-5	3	5
9_2	2	2	2	1-3	2	2-3	2-3	2-3	3	2-3	2	2-3	2	2-3	2
9_2^*	3	2	2	2-3	2	3	2-3	2-3	1-3	2-3	2	2-3	2	2-3	2
9_3	5	4	3-4	4-5	3-4	5	3-5	4-5	1-5	4-5	2-3	3	3-4	2-3	4
9_3^*	2	2-3	3-4	2-4	3-4	2-3	3-5	2-5	5	2-4	4	3-4	3-4	4-5	2-3
9_4	2	2	2-3	2-4	2-3	2-3	2-4	2-4	4	2-4	3	2-4	2-3	3-4	2-3
9_4^*	4	3	2-3	3-4	2-3	4	2-4	3-4	1-4	3-4	2-3	2-3	2-3	1-3	3
9_5	3-4	2-3	2-3	2-4	2-3	3-4	2-4	2-4	1-4	2-4	2-3	2	2-3	1-2	2-3
9_5^*	2	2	2-3	1-3	2-3	2-3	2-4	2-4	3-4	2-3	2-3	2-3	2-3	2-4	2
9_6	1	2-3	3-4	2-3	3-4	2	3-5	2-4	5	2-3	4	3	3-4	4	2-3
9_6^*	5	4	3	4-5	3-4	5	3-5	4-5	2-4	4-5	2	3	3-4	2	4
9_7	1	2	2-3	2	2-3	2	2-4	2-3	4	2	3	2	2-3	3	2-3
9_7^*	4	3	2	3-4	2-3	4	2-4	3-4	1-3	3-4	2	2	2-3	1-3	3
9_8	2	1	2	2	2-3	2	2	2	3	2	2-3	2	2-3	2-3	2-3
9_8^*	3-4	2	2	2	2	3	2	2	1-3	2	2	2	2-3	1-3	2
9_9	1	2-3	3-4	2-3	3-4	2-3	3-5	2-4	5	2-3	4	3	3-4	4	2-3

Table C.1: Continued.

New Knot Distance Table (cont.)

	8_{12}	8_{13}	8_{14}	8_{15}	8_{16}	8_{17}	8_{18}	8_{19}	8_{20}	8_{21}	$3_1 \# 5_1$	$3_1^* \# 5_1$	$3_1 \# 5_2$
8_{10}	2-4	2	2-3	3-4	2	1-3	1-3	2	1-2	2-3	4	2-4	3
8_{10}^*	2-4	2	1-2	1-2	1-2	1-3	1-3	4	1-2	1-3	2	2	1-2
8_{11}	2	2	2	2	1-2	1-2	1-3	4	1-2	2	2-3	2-3	2
8_{11}^*	2	2	2	3	2-3	1-2	1-3	2-3	1-2	2	4	2-4	3
8_{12}	0	2	2	2-4	1-2	2-3	2-4	3-5	1-3	2-3	3-5	2-5	2-4
8_{13}	2	0	2	2	1-2	2	2-3	3-4	1-2	2	3	2-3	2
8_{13}^*	2	2	2	2-3	1-2	2	2-3	3	1-2	2	3-4	2-4	2-3
8_{14}	2	2	0	1-2	1-2	1-2	2	4	1-2	1-2	2-3	1-3	1-2
8_{14}^*	2	2	2	3	2	1-2	2	2-3	1-2	2	4	2-3	3
8_{15}	2-4	2	1-2	0	1-2	2-3	2-3	5	2	1-2	2	2	1-2
8_{15}^*	2-4	2-3	3	4	3-4	2-3	2-3	1-2	2-3	3	5	3-4	4
8_{16}	1-2	1-2	1-2	1-2	0	1-2	2	4	1-2	1-3	2-3	1-3	1-2
8_{16}^*	1-2	1-2	2	3-4	2	1-2	2	2-3	1-2	2-3	4	2-3	3
8_{17}	2-3	2	1-2	2-3	1-2	0	2	3	1-2	1-2	3	2-3	2
8_{18}	2-4	2-3	2	2-3	2	2	0	3	1-3	1-3	3	1-3	2
8_{19}	3-5	3-4	4	5	4	3	3	0	3-4	4	6	4	5
8_{19}^*	3-5	3	2-3	1-2	2-3	3	3	6	3	2	2	2	1-2
8_{20}	1-3	1-2	1-2	2	1-2	1-2	1-3	3-4	0	1-2	3	1-3	2
8_{20}^*	1-3	1-2	1-2	2-3	1-2	1-2	1-3	3	1-2	1-2	3-4	1-4	2-3
8_{21}	2-3	2	1-2	1-2	1-3	1-2	1-3	4	1-2	0	2	1-4	1-2
8_{21}^*	2-3	2	2	3	2-3	1-2	1-3	2	1-2	2	4	2-4	3
$3_1 \# 5_1$	3-5	3	2-3	2	2-3	3	3	6	3	2	0	2	1
$3_1^* \# 5_1^*$	3-5	3-4	4	5	4	3	3	2	3-4	4	6	4	5
$3_1 \# 5_1^*$	2-5	2-4	2-3	3-4	2-3	2-3	1-3	2	1-2	2-4	4	2-4	3
$3_1^* \# 5_1$	2-5	2-3	1-3	2	1-3	2-3	1-3	4	1-3	1-4	2	0	1-3
$3_1 \# 5_2$	2-4	2	1-2	1-2	1-2	2	2	5	2	1-2	1	1-3	0
$3_1^* \# 5_2^*$	2-4	2-3	3	4	3	2	2	1-2	2-3	3	5	3	4
$3_1 \# 5_2^*$	2-4	2-3	2	2-3	1-2	1-2	1-2	3	1-2	1-3	3	1-3	2
$3_1^* \# 5_2$	2-4	2	2	2	1-2	1-2	1-2	3	1-2	1-3	3	1	2
$4_1 \# 4_1$	2	2	2	2-4	1-2	2-3	2-4	3-5	2-3	1-3	3-5	1-5	2-4
9_1	4-6	4	3-4	2-3	3-4	4	4	7	4	3-5	2-3	3	2-4
9_1^*	4-6	4-5	5	6	5	4	4	2-3	4-5	5	7	5	6
9_2	2-3	2	2	2-3	1-3	2	1-3	4	1-2	1-2	2-4	1-4	1-3
9_2^*	2-3	2	2	3	2-3	2	1-3	2-4	1-2	2	4	2-4	3
9_3	3-5	3-4	4	5	4-5	3-4	3-4	1-3	3-4	4	6	4-5	5
9_3^*	3-5	3	2-3	1-3	2-3	3-4	3-4	6	3	2-4	1-3	2-3	1-3
9_4	2-4	2-3	2-3	1-3	1-3	2-3	2-4	5	2-3	1-3	1-3	1-3	1-3
9_4^*	2-4	2-3	3	4	3-4	2-3	2-4	1-3	2-3	3	5	3-5	4
9_5	2-4	2-3	2-3	3-4	2-4	1-3	2-3	2-3	1-3	2-3	4-5	2-4	3-4
9_5^*	2-4	2	2	1-2	1-2	1-3	2-3	4-5	1-2	1-3	2-3	2-3	1-2
9_6	3-5	3	2-3	2	2-3	3	3	6	3	2-4	2	2	2-3
9_6^*	3-5	3-4	4	5	4	3	3	2	3-4	4	6	4	5
9_7	2-4	2-3	2	2-3	2	2	2	5	2-3	1-3	1-3	1-3	1-2
9_7^*	2-4	2-3	3	4	3	2	2	1-3	2-3	3	5	3	4
9_8	2	2	2	1-3	1-2	1-2	2	4-5	1-3	1-3	2-3	1-3	1-2
9_8^*	2	2	2	3-4	2	1-2	2	2-3	1-3	2-3	4-5	2-3	3-4
9_9	3-5	3	2-3	1-3	2-3	3	3	6	3	2-4	1-3	2-3	1-3

Table C.1: Continued.

New Knot Distance Table (cont.)

	$3_1^* \# 5_2$	$4_1 \# 4_1$	9_1	9_2	9_3	9_4	9_5	9_6	9_7	9_8	9_9	9_{10}	9_{11}	9_{12}
8_{10}	1-3	2-4	5	2-3	2-3	3-4	1-2	4	3	2-3	4	1-3	1-2	2-3
8_{10}^*	1-2	2-4	3	2-3	4-5	1-3	2-4	2	1-3	1-3	2-3	3-4	3-4	2-3
8_{11}	2	2-3	3-4	2	4	2-3	2-3	2-3	2-3	2-3	2-3	3-4	3	2
8_{11}^*	2-3	2-3	5	2	2-3	3	2	4	3	2	4	2-3	2	2
8_{12}	2-4	2	4-6	2-3	3-5	2-4	2-4	3-5	2-4	2	3-5	2-5	2-3	2
8_{13}	2	2	4	2	3-4	2-3	2-3	3	2-3	2	3	2-4	2-3	2
8_{13}^*	2-3	2	4-5	2	3	2-3	2	3-4	2-3	2	3-4	2-3	2	2
8_{14}	2	2	3-4	2	4	2-3	2-3	2-3	2	2	2-3	3-4	3	2
8_{14}^*	2	2	5	2	2-3	3	2	4	3	2	4	2-3	2	2
8_{15}	2	2-4	2-3	2-3	5	1-3	3-4	2	2-3	1-3	1-3	4-5	4	2-3
8_{15}^*	2-3	2-4	6	3	1-3	4	1-2	5	4	3-4	5	1-3	1-2	3
8_{16}	1-2	1-2	3-4	1-3	4-5	1-3	2-4	2-3	2	1-2	2-3	3-4	3	1-3
8_{16}^*	1-2	1-2	5	2-3	2-3	3-4	1-2	4	3	2	4	2-3	2	2-3
8_{17}	1-2	2-3	4	2	3-4	2-3	1-3	3	2	1-2	3	2-3	2-3	2
8_{18}	1-2	2-4	4	1-3	3-4	2-4	2-3	3	2	2	3	2-3	2-3	2-3
8_{19}	3	3-5	7	4	1-3	5	2-3	6	5	4-5	6	1-3	1-2	4
8_{19}^*	3	3-5	2-3	2-4	6	1-3	4-5	2	1-3	2-3	1-3	5-6	5	2-4
8_{20}	1-2	2-3	4	1-2	3-4	2-3	1-3	3	2-3	1-3	3	2-4	2-3	1-2
8_{20}^*	1-3	2-3	4-5	1-2	3	2-3	1-2	3-4	2-3	1-3	3-4	2-3	2	1-2
8_{21}	1-3	1-3	3-5	1-2	4	1-3	2-3	2-4	1-3	1-3	2-4	3-4	3	1-2
8_{21}^*	1-3	1-3	5	2	2-4	3	1-3	4	3	2-3	4	1-4	1-3	2
$3_1 \# 5_1$	3	3-5	2-3	2-4	6	1-3	4-5	2	1-3	2-3	1-3	5-6	5	2-4
$3_1^* \# 5_1^*$	3	3-5	7	4	1-3	5	2-3	6	5	4-5	6	2-3	2	4
$3_1 \# 5_1^*$	1-3	1-5	5	2-4	2-3	3-5	2-3	4	3	2-3	4	2-3	2	2-4
$3_1^* \# 5_1$	1	1-5	3	1-4	4-5	1-3	2-4	2	1-3	1-3	2-3	3-4	3-4	1-4
$3_1 \# 5_2$	2	2-4	2-4	1-3	5	1-3	3-4	2-3	1-2	1-2	1-3	4-5	4	1-3
$3_1^* \# 5_2^*$	2	2-4	6	3	1-3	4	1-2	5	4	3-4	5	1-3	1-2	3
$3_1 \# 5_2^*$	1-3	2-4	4	1-3	3	2-4	2	3	2	2	3	2-3	2	1-3
$3_1^* \# 5_2$	0	2-4	4	1-3	3-4	2-3	2-3	3	2-3	2-3	3	2-3	2-3	1-3
$4_1 \# 4_1$	2-4	0	4-6	1-3	3-5	2-4	2-4	3-5	2-4	2	3-5	2-5	2-3	1-3
9_1	4	4-6	0	3	7	2	5-6	2-1	2	3-4	2-1	6-7	6	3-4
9_1^*	4	4-6	8	5	2-1	6	3	7	6	5-6	7	2	2-3	5
9_2	1-3	1-3	3	0	4	1	2-3	2	1	2	2	3-4	3	1
9_2^*	1-3	1-3	5	2	2	3	2	4	3	2-3	4	2-3	2	2
9_3	3-4	3-5	7	4	0	5	2	6	5	4-5	6	1	2	4
9_3^*	3	3-5	2-1	2	6	1	4-5	2	2-3	2-4	2	5-6	5	2-3
9_4	2-3	2-4	2	1	5	0	3-4	2-3	2	2-3	1	4-5	4	2
9_4^*	2-4	2-4	6	3	1	4	1	5	4	3-4	5	2	1	3
9_5	2-3	2-4	5-6	2-3	2	3-4	0	4-5	3-4	2-4	4-5	1	2	2-3
9_5^*	2	2-4	3	2	4-5	1	2-4	2-3	2	2-3	2	3-5	3-4	2
9_6	3	3-5	2-1	2	6	2-3	4-5	0	1	2-3	2	5-6	5	2-3
9_6^*	3	3-5	7	4	2	5	2-3	6	5	4-5	6	2-3	2	4
9_7	2-3	2-4	2	1	5	2	3-4	1	0	2	1	4-5	4	2
9_7^*	2	2-4	6	3	2-3	4	2	5	4	3-4	5	2-3	2-3	3
9_8	2-3	2	3-4	2	4-5	2-3	2-4	2-3	2	0	2-3	3-5	3	1
9_8^*	2	2	5-6	2-3	2-4	3-4	2-3	4-5	3-4	2	4-5	2-3	2	2
9_9	3	3-5	2-1	2	6	1	4-5	2	1	2-3	0	5-6	5	2-3

Table C.1: Continued.

New Knot Distance Table (cont.)

	0_1	3_1	4_1	5_1	5_2	6_1	6_2	6_3	$3_1 \# 3_1$	$3_1 \# 3_1^*$	7_1	7_2	7_3
9_9^*	3	4	3-4	5	4	3-4	4	3	5	3	6	4	1
9_{10}	2-3	3-4	2-4	4-5	3-4	2-4	3-4	2-3	4-5	2-3	5-6	3-4	1
9_{10}^*	2-3	2	2-4	2	2	2-4	2-3	2-3	1-3	2-3	2	2	4-5
9_{11}	2	3	2	4	3	2-3	3	2-3	4	2-3	5	3	1
9_{11}^*	2	2	2	1	1	2-3	2-3	2-3	1-3	2-3	2	2	4
9_{12}	1	2	2	2-3	2	1	2	2	2-3	2-3	2-3	1	3
9_{12}^*	1	2	2	3	2	2	2	2	3	2-3	4	2	2
9_{13}	2-3	3-4	2-4	4-5	3-4	2-4	3-4	2-3	4-5	2-3	5-6	3-4	1
9_{13}^*	2-3	2	2-4	1	2	2-4	2-3	2-3	2-3	2-3	2	2	4-5
9_{14}	1	2	2	2-3	2	2	2	2	2-3	2	3-4	2	2
9_{14}^*	1	1	2	2	1	1	2	2	2	2	3	2	2-3
9_{15}	2	2-3	1	3-4	2-3	2	2	2-3	3-4	2-3	4-5	2-3	2
9_{15}^*	2	2	1	2	1	2	2	2-3	2-3	2-3	2-3	1	3-4
9_{16}	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	3-8	1-8	2-8
9_{16}^*	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	3-8	1-8	2-8
9_{17}	2	2	1	2-3	2-3	2	1	2-3	2-3	2-3	2-4	2-3	3-4
9_{17}^*	2	2	1	3	2-3	2	2	2-3	3	2-3	4	2-3	2-4
9_{18}	2	2	2-3	2	1	2-3	2-3	2-3	2-3	2-3	2	1	4
9_{18}^*	2	3	2-3	4	3	2-3	3	2-3	4	2-3	5	3	1
9_{19}	1	2	1	2-3	2	1	2	2	2-3	2-3	3-4	2	2-3
9_{19}^*	1	2	1	2-3	2	2	2	2	2-3	2-3	3-4	2	2-3
9_{20}	2	1	2	2	1	2	1	2	2	2	2	2	4
9_{20}^*	2	3	2	4	3	2-3	3	2	4	2	5	3	2
9_{21}	1	2	2	3	2	2	2	2	3	2	4	2	2
9_{21}^*	1	1	2	2	1	2	2	2	2	2	2-3	2	3
9_{22}	1-8	1-8	2-8	2-8	1-8	1-8	1-8	1-8	2-8	2-8	3-8	1-8	2-8
9_{22}^*	1-8	1-8	2-8	2-8	1-8	1-8	1-8	1-8	2-8	2-8	3-8	1-8	2-8
9_{23}	2	2	2-3	1	1	2-3	2-3	2-3	2-3	2-3	2	2	4
9_{23}^*	2	3	2-3	4	3	2-3	3	2-3	4	2-3	5	3	2
9_{24}	1-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8	2-8	1-8	3-8	1-8	2-8
9_{24}^*	1-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8	2-8	1-8	3-8	1-8	2-8
9_{25}	1-8	1-8	2-8	2-8	1-8	1-8	1-8	2-8	2-8	2-8	3-8	1-8	2-8
9_{25}^*	1-8	1-8	2-8	2-8	1-8	1-8	1-8	2-8	2-8	2-8	3-8	1-8	2-8
9_{26}	1	2	2	3	2	2	2	2	3	2	4	2	2
9_{26}^*	1	1	2	1	1	2	1	2	2	2	2	2	3
9_{27}	1	1	1	2	2	2	2	1	2	2	3	2	2-3
9_{27}^*	1	2	1	2-3	2	2	1	1	2-3	2	3-4	2	2-3
9_{28}	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	3-8	1-8	2-8
9_{28}^*	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	3-8	1-8	2-8
9_{29}	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	3-8	1-8	2-8
9_{29}^*	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	3-8	1-8	2-8
9_{30}	1-8	1-8	2-8	2-8	1-8	1-8	1-8	1-8	2-8	2-8	3-8	1-8	2-8
9_{30}^*	1-8	1-8	2-8	2-8	1-8	1-8	1-8	1-8	2-8	2-8	3-8	1-8	2-8
9_{31}	2	1	2-3	1	1	2-3	2	1	2	2	2	2	3-4
9_{31}^*	2	2	2-3	3	2-3	2-3	2-3	1	3	2	4	2-3	2
9_{32}	1-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8	2-8	2-8	3-8	1-8	2-8
9_{32}^*	1-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8	2-8	2-8	3-8	1-8	2-8

Table C.1: Continued.

New Knot Distance Table (cont.)

	7_4	7_5	7_6	7_7	$3_1 \# 4_1$	8_1	8_2	8_3	8_4	8_5	8_6	8_7	8_8	8_9
9_9^*	2	5	4	3	4-5	3-4	5	3-5	4-5	1-4	4-5	2-3	3	3-4
9_{10}	1	4-5	3-4	2-3	3-5	2-4	4-5	2-5	3-5	1-4	3-5	2-3	2-3	2-4
9_{10}^*	3-5	2-3	2-3	2-4	1-3	2-4	2-3	2-5	2-4	4-5	2-3	3-4	2-3	2-4
9_{11}	2	4	3	2-3	3	2-3	4	2-4	3	1-4	3-4	2	2	2-3
9_{11}^*	3-4	2	1	2-3	1-3	2-3	2	2-4	2-3	4	2-3	3	2-3	2-3
9_{12}	2-3	2	1	2	1-3	2	2-3	2	2-3	3	2	2	2-3	2
9_{12}^*	2-3	3	2	2	2-3	2	3	2	2	1-3	2-3	2	2-3	2
9_{13}	1	4-5	3-4	2-3	3-5	2-4	4-5	2-5	3-5	2-4	3-5	2	2-3	2-4
9_{13}^*	3-5	1	2-3	2-4	1-3	2-4	2	2-5	2-4	4-5	2-3	3-4	2-3	2-4
9_{14}	2	2-3	2	1	2-3	2	2-3	2	2	2-3	2-3	2	2	2
9_{14}^*	2-3	2	2	2	2	2	2	2	2-3	2-3	2	2	2	2
9_{15}	2	3-4	2	2	2	2-3	3	2-3	2	1-3	2-3	2-3	2	2-3
9_{15}^*	2-4	2	1	2	2	2-3	2-3	2-3	2	3	2-3	2-3	2-3	2-3
9_{16}	2-8	2-8	1-8	1-8	2-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{16}^*	2-8	2-8	1-8	1-8	2-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{17}	2-3	2-3	2	1	2	2-3	2	2-3	2	3	2	2-3	2-3	2
9_{17}^*	2-3	3	2	2	2	2-3	3	2-3	2	1-2	2-3	2-3	2-3	2
9_{18}	3-4	1	2	2-3	2-3	2-3	2-3	2-4	2-4	4	2-3	3	2-3	2-3
9_{18}^*	2	4	3	2-3	3-4	2-3	4	2-4	3-4	2-4	3-4	2-3	2	2-3
9_{19}	2-3	2-3	2	1	2	2	2-3	2	2	2-3	2	2	2-3	2
9_{19}^*	2-3	2-3	2	2	2	2	2-3	2	2	2-3	2-3	2	2-3	2
9_{20}	3-4	1	1	2-3	2	2-3	2	2-3	2	4	2	3	2	2
9_{20}^*	2	4	3	2	3	2-3	4	2-3	3	2	3-4	2	2	2
9_{21}	1	3	2	2	2-3	2	3	2-3	2-3	1-3	2-3	2	2	2
9_{21}^*	2-3	2	1	2	1-2	2	2	2-3	2-3	3	2	2	2	2
9_{22}	1-8	2-8	1-8	1-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8
9_{22}^*	1-8	2-8	1-8	1-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8
9_{23}	3-4	1	2	2-3	2-3	2-3	2	2-4	2-4	4	2-3	3	2-3	2-3
9_{23}^*	2	4	3	2-3	3-4	2-3	4	2-4	3-4	2-4	3-4	2	2	2-3
9_{24}	1-8	2-8	1-8	1-8	1-8	2-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	2-8
9_{24}^*	1-8	2-8	1-8	1-8	1-8	2-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	2-8
9_{25}	1-8	2-8	1-8	1-8	1-8	2-8	2-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8
9_{25}^*	1-8	2-8	1-8	1-8	1-8	2-8	2-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8
9_{26}	2	3	2	1	2-3	2	3	2-3	2-3	1-2	2-3	2	2	2
9_{26}^*	2-3	2	2	2	1-2	2	2	2-3	2	3	2	2	2	2
9_{27}	2-3	2	1	2	2	2	2	2-3	2	2	2	2	2	2
9_{27}^*	2	2-3	2	2	2	2	2	2-3	2	2-3	2	2	2	2
9_{28}	2-8	2-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{28}^*	2-8	2-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{29}	2-8	2-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{29}^*	2-8	2-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{30}	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8
9_{30}^*	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8	1-8	1-8	2-8
9_{31}	2-3	2	2	2-3	1-2	2-3	2	2-4	2-3	3-4	2	2	2	2-3
9_{31}^*	2	3	2-3	2	2-3	2-3	3	2-4	2-4	2-3	2-3	2	2	2-3
9_{32}	2-8	2-8	1-8	1-8	2-8	2-8	2-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{32}^*	2-8	2-8	1-8	1-8	2-8	2-8	2-8	1-8	1-8	2-8	1-8	1-8	2-8	1-8

Table C.1: Continued.

New Knot Distance Table (cont.)

	8_{10}	8_{11}	8_{12}	8_{13}	8_{14}	8_{15}	8_{16}	8_{17}	8_{18}	8_{19}	8_{20}	8_{21}	$3_1 \# 5_1$	$3_1^* \# 5_1$
9_9^*	2-3	4	3-5	3-4	4	5	4	3	3	1-3	3-4	4	6	4
9_{10}	1-3	3-4	2-5	2-4	3-4	4-5	3-4	2-3	2-3	1-3	2-4	3-4	5-6	3-4
9_{10}^*	3-4	2-3	2-5	2-3	2-3	1-3	2-3	2-3	2-3	5-6	2-3	1-4	2-3	2-3
9_{11}	1-2	3	2-3	2-3	3	4	3	2-3	2-3	1-2	2-3	3	5	3-4
9_{11}^*	3-4	2	2-3	2	2	1-2	2	2-3	2-3	5	2	1-3	2	2
9_{12}	2-3	2	2	2	2	2-3	1-3	2	2-3	4	1-2	1-2	2-4	1-4
9_{12}^*	2-3	2	2	2	2	3	2-3	2	2-3	2-4	1-2	2	4	2-4
9_{13}	2	3-4	2-5	2-4	3-4	4-5	3-4	2-3	2-3	2	2-4	3-4	5-6	3-4
9_{13}^*	3-4	2-3	2-5	2-3	2-3	2	1-3	2-3	2-3	5-6	2-3	1-4	2	2
9_{14}	1-2	2	2	2	2	2-3	2-3	1-2	2	3	1-2	2	3-4	2-3
9_{14}^*	1-3	2	2	2	2	2	2	1-2	2	3-4	1-2	2	3	2-3
9_{15}	2	2-3	2	2	2	3-4	2	1-3	1-3	2-3	2-3	2-3	4-5	2-4
9_{15}^*	2-4	2	2	2	2	1-2	2	1-3	1-3	4-5	2	1-3	2-3	1-3
9_{16}	1-8	2-8	2-8	2-8	1-8	2-8	1-8	1-8	1-8	3-8	1-8	1-8	3-8	1-8
9_{16}^*	1-8	2-8	2-8	2-8	1-8	2-8	1-8	1-8	1-8	3-8	1-8	1-8	3-8	1-8
9_{17}	2-4	2	2	2	2	1-4	2	1-3	1-3	4	2-3	1-2	2-4	1-4
9_{17}^*	2-4	2-3	2	2	2	3-4	2	1-2	1-3	2-4	2-3	2-3	4	2-4
9_{18}	3-4	2	2-4	2	2	2	1-2	2-3	2-3	5	2	1-3	1-3	1-3
9_{18}^*	1-2	3	2-4	2-3	3	4	3-4	2-3	2-3	1-3	2-3	3	5	3-4
9_{19}	2-3	2	2	2	2	2-3	2	1-2	2-3	3-4	1-2	2	3-4	2-4
9_{19}^*	2-3	2	2	2	2	2-3	2	1-2	2-3	3-4	1-2	2	3-4	2-4
9_{20}	3	2	2-3	2	2	2	1-2	2	2	5	2	1-2	1-3	1-3
9_{20}^*	1-2	3	2-3	2-3	3	4	3	2	2	1-3	2-3	3	5	3
9_{21}	1-2	2	2-3	2	2	3	2-3	1-2	2	2-3	1-2	2	4	2-3
9_{21}^*	2-3	2	2-3	2	2	1-2	1-2	1-2	2	4	1-2	1-2	2-3	2-3
9_{22}	1-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	3-8	1-8	1-8	3-8	2-8
9_{22}^*	1-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	3-8	1-8	1-8	3-8	2-8
9_{23}	3-4	2	2-4	2	2	2	1-2	2-3	2-3	5	2	2-3	2	2
9_{23}^*	1-2	3	2-4	2-3	3	4	3-4	2-3	2-3	2	2-3	3	5	3-4
9_{24}	1-8	1-8	1-8	1-8	1-8	2-8	1-8	1-8	1-8	3-8	1-8	1-8	3-8	1-8
9_{24}^*	1-8	1-8	1-8	1-8	1-8	2-8	1-8	1-8	1-8	3-8	1-8	1-8	3-8	1-8
9_{25}	1-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	3-8	1-8	1-8	3-8	2-8
9_{25}^*	1-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	3-8	1-8	1-8	3-8	2-8
9_{26}	1-2	2	2-3	2	2	3	2-3	1-2	2	2	1-2	2	4	2-3
9_{26}^*	2-3	2	2-3	2	2	1-2	1-2	1-2	2	4	1-2	1-2	2	2
9_{27}	1-2	2	2	2	2	2-3	2	1-2	2	3-4	1-2	1-2	3	1-3
9_{27}^*	1-2	2	2	2	2	2-3	2	1-2	2	3	1-2	1-2	3-4	1-3
9_{28}	1-8	2-8	1-8	1-8	1-8	2-8	1-8	1-8	1-8	3-8	1-8	1-8	3-8	1-8
9_{28}^*	1-8	2-8	1-8	1-8	1-8	2-8	1-8	1-8	1-8	3-8	1-8	1-8	3-8	1-8
9_{29}	1-8	2-8	1-8	1-8	1-8	2-8	1-8	1-8	1-8	3-8	1-8	1-8	3-8	1-8
9_{29}^*	1-8	2-8	1-8	1-8	1-8	2-8	1-8	1-8	1-8	3-8	1-8	1-8	3-8	1-8
9_{30}	1-8	1-8	1-8	1-8	1-8	2-8	1-8	1-8	2-8	3-8	1-8	1-8	3-8	2-8
9_{30}^*	1-8	1-8	1-8	1-8	1-8	2-8	1-8	1-8	2-8	3-8	1-8	1-8	3-8	2-8
9_{31}	2	2	2-4	2	2	1-2	1-2	1-2	2	4	2	1-3	2	1-2
9_{31}^*	1-2	2-3	2-4	2	2-3	3-4	2	1-2	2	2	2	2-3	4	2-3
9_{32}	1-8	1-8	1-8	1-8	1-8	2-8	1-8	1-8	2-8	3-8	1-8	1-8	3-8	1-8
9_{32}^*	1-8	1-8	1-8	1-8	1-8	2-8	1-8	1-8	2-8	3-8	1-8	1-8	3-8	1-8

Table C.1: Continued.

New Knot Distance Table (cont.)

	$3_1 \# 5_2$	$3_1^* \# 5_2$	$4_1 \# 4_1$	9_1	9_2	9_3	9_4	9_5	9_6	9_7	9_8	9_9
9_9^*	5	3	3-5	7	4	2	5	2	6	5	4-5	6
9_{10}	4-5	2-3	2-5	6-7	3-4	1	4-5	1	5-6	4-5	3-5	5-6
9_{10}^*	1-3	2-3	2-5	2	2-3	5-6	2	3-5	2-3	2-3	2-3	2
9_{11}	4	2-3	2-3	6	3	2	4	2	5	4	3	5
9_{11}^*	1-2	2	2-3	2-3	2	5	1	3-4	2	2-3	2	2
9_{12}	1-3	1-3	1-3	3-4	1	4	2	2-3	2-3	2	1	2-3
9_{12}^*	3	1-3	1-3	5	2	2-3	3	2	4	3	2	4
9_{13}	4-5	2-3	2-5	6-7	3-4	1	4-5	1	5-6	4-5	3-5	5-6
9_{13}^*	2-3	2-3	2-5	2	2-3	5-6	2	3-5	2	2	2-3	1
9_{14}	2-3	1-2	2-3	4-5	2	3	2-3	2	3-4	2-3	2	3-4
9_{14}^*	2	1-2	2-3	4	2	3-4	2-3	2-3	3	2	2	3
9_{15}	3-4	1-3	2	5-6	2-3	2-3	3-4	2	4-5	3-4	2	4-5
9_{15}^*	1-2	1-2	2	3-4	1	4-5	2	2-4	2-3	2	1	2-3
9_{16}	2-8	1-8	2-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{16}^*	2-8	1-8	2-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{17}	1-3	1-3	2	3-5	2-3	4-5	2-4	2-4	2-4	2-3	2	2-4
9_{17}^*	3	1-3	2	5	2-3	2-5	3-4	2-4	4	3	2	4
9_{18}	2	2	2-4	2	1	5	2	3-4	1	2	2-3	2
9_{18}^*	4	2-3	2-4	6	3	1	4	2	5	4	3-4	5
9_{19}	2-3	1-3	2	4-5	2	3-4	2-3	2-3	3-4	2-3	2	3-4
9_{19}^*	2-3	1-3	2	4-5	2	3-4	2-3	2-3	3-4	2-3	2	3-4
9_{20}	2	2	2-3	2-3	2	5	2-3	3-4	2	1	2	2
9_{20}^*	4	2	2-3	6	3	2-3	4	2	5	4	3	5
9_{21}	3	1-2	2-3	5	2	2-3	3	1	4	3	2-3	4
9_{21}^*	2	1-2	2-3	3-4	2	4	2	2-3	2-3	2	2	2-3
9_{22}	2-8	1-8	2-8	4-8	2-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{22}^*	2-8	1-8	2-8	4-8	2-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{23}	2	2	2-4	2	2-3	5	2-3	3-4	1	2	2-3	2
9_{23}^*	4	2-3	2-4	6	3	2-3	4	2	5	4	3-4	5
9_{24}	2-8	1-8	1-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{24}^*	2-8	1-8	1-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{25}	2-8	1-8	2-8	4-8	2-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{25}^*	2-8	1-8	2-8	4-8	2-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{26}	3	1-2	2-3	5	2	2-3	3	2	4	3	2-3	4
9_{26}^*	2	1-2	2-3	3	2	4	2-3	2-3	2	2	2	2-3
9_{27}	2	2-3	2	4	2	3-4	2-3	2-3	3	2	1	3
9_{27}^*	2-3	2	2	4-5	2	3-4	2-3	2-3	3-4	2-3	2	3-4
9_{28}	2-8	1-8	2-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{28}^*	2-8	1-8	2-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{29}	2-8	1-8	2-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{29}^*	2-8	1-8	2-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{30}	2-8	2-8	2-8	4-8	2-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{30}^*	2-8	2-8	2-8	4-8	2-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{31}	2	2	1-4	3	2-3	4-5	2-3	2-4	2	2	2	2-3
9_{31}^*	3	2	1-4	5	2-3	2-3	3-4	2	4	3	2-3	4
9_{32}	2-8	1-8	2-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8
9_{32}^*	2-8	1-8	2-8	4-8	1-8	3-8	2-8	1-8	3-8	2-8	1-8	3-8

Table C.1: Continued.

New Knot Distance Table (cont.)

	9_{10}	9_{11}	9_{12}	9_{13}	9_{14}	9_{15}	9_{16}	9_{17}	9_{18}	9_{19}	9_{20}	9_{21}	9_{22}	9_{23}	9_{24}
9_9^*	2	2	4	1	3	2-3	3-8	4	5	3-4	5	2-3	3-8	5	3-8
9_{10}	0	2	3-4	2	2-3	2-3	2-8	3-4	4-5	2-4	4-5	2	2-8	4-5	2-8
9_{10}^*	4-6	4-5	2-3	4-6	2-4	3-5	2-8	2-4	2	2-4	2-3	3-4	2-8	2-3	2-8
9_{11}	2	0	3	2	2	1	2-8	3	4	2-3	4	1	2-8	4	2-8
9_{11}^*	4-5	4	2	4-5	2-3	3	2-8	2-3	2	2-3	2	3	2-8	2	2-8
9_{12}	3-4	3	0	3-4	2	2-3	1-8	2-3	2	2	1	2	2-8	2-3	1-8
9_{12}^*	2-3	2	2	2-3	2	2	1-8	2-3	3	2	3	2	2-8	3	1-8
9_{13}	2	2	3-4	0	2-3	2-3	2-8	3-4	4-5	2-4	4-5	2	2-8	4-5	2-8
9_{13}^*	4-6	4-5	2-3	4-6	2-4	3-5	2-8	2-4	2	2-4	2	3-4	2-8	2	2-8
9_{14}	2-3	2	2	2-3	0	2	2-8	2	2-3	1	2-3	2	2-8	2-3	2-8
9_{14}^*	2-4	2-3	2	2-4	2	2-3	2-8	2-3	2	2	2	2	2-8	2	2-8
9_{15}	2-3	1	2-3	2-3	2	0	1-8	2	3-4	2	3	2	2-8	3-4	1-8
9_{15}^*	3-5	3	2	3-5	2-3	2	1-8	2	2	2	2	2-3	2-8	2	1-8
9_{16}	2-8	2-8	1-8	2-8	2-8	1-8	0	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{16}^*	2-8	2-8	1-8	2-8	2-8	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{17}	3-4	3	2-3	3-4	2	2	1-8	0	2-4	1	2	2-3	2-8	2-4	1-8
9_{17}^*	2-4	2-3	2-3	2-4	2-3	2	1-8	2	3-4	2	3	2-3	2-8	3-4	1-8
9_{18}	4-5	4	2	4-5	2-3	3-4	2-8	2-4	0	2-3	2	3	2-8	2	2-8
9_{18}^*	2	2	3	2	2	2	2-8	3-4	4	2-3	4	2	2-8	4	2-8
9_{19}	2-4	2-3	2	2-4	1	2	1-8	1	2-3	0	2-3	2	1-8	2-3	1-8
9_{19}^*	2-4	2-3	2	2-4	2	2	1-8	2	2-3	2	2-3	2	1-8	2-3	1-8
9_{20}	4-5	4	1	4-5	2-3	3	2-8	2	2	2-3	0	3	2-8	2	2-8
9_{20}^*	2-3	2	3	2	2	2	2-8	3	4	2-3	4	2	2-8	4	2-8
9_{21}	2	1	2	2	2	2	1-8	2-3	3	2	3	0	1-8	3	1-8
9_{21}^*	3-4	3	2	3-4	2	2-3	1-8	2-3	2	2	2	2	1-8	2	1-8
9_{22}	2-8	2-8	2-8	2-8	2-8	2-8	1-8	2-8	2-8	1-8	2-8	1-8	0	2-8	1-8
9_{22}^*	2-8	2-8	2-8	2-8	2-8	2-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{23}	4-5	4	2-3	4-5	2-3	3-4	2-8	2-4	2	2-3	2	3	2-8	0	2-8
9_{23}^*	2-3	2	3	2	2	2	2-8	3-4	4	2-3	4	2	2-8	4	2-8
9_{24}	2-8	2-8	1-8	2-8	2-8	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	0
9_{24}^*	2-8	2-8	1-8	2-8	2-8	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{25}	2-8	2-8	2-8	2-8	1-8	1-8	1-8	1-8	2-8	2-8	2-8	1-8	1-8	2-8	1-8
9_{25}^*	2-8	2-8	2-8	2-8	1-8	1-8	1-8	1-8	2-8	2-8	2-8	1-8	1-8	2-8	1-8
9_{26}	2-3	2	2	2	1	2	1-8	2	3	2	3	2	1-8	3	1-8
9_{26}^*	3-4	3	2	3-4	2	2-3	1-8	2	2	2	2	2	1-8	2	1-8
9_{27}	2-4	2-3	2	2-4	2	2	1-8	2	2-3	2	2	2	1-8	2-3	1-8
9_{27}^*	2-3	2	2	2-3	2	2	1-8	2	2-3	2	2	2	1-8	2-3	1-8
9_{28}	2-8	2-8	1-8	2-8	2-8	2-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{28}^*	2-8	2-8	1-8	2-8	2-8	2-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{29}	2-8	2-8	1-8	2-8	2-8	2-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{29}^*	2-8	2-8	1-8	2-8	2-8	2-8	1-8	2-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{30}	2-8	2-8	2-8	2-8	1-8	2-8	1-8	2-8	2-8	2-8	2-8	1-8	1-8	2-8	1-8
9_{30}^*	2-8	2-8	2-8	2-8	1-8	2-8	1-8	2-8	2-8	2-8	2-8	1-8	1-8	2-8	1-8
9_{31}	3-4	3-4	2-3	3-4	2-3	2-4	2-8	2-3	2	2-3	2	2-3	1-8	2	1-8
9_{31}^*	2-3	2	2-3	2	2	2	2-8	2-3	3-4	2-3	3	2	1-8	3-4	1-8
9_{32}	2-8	2-8	1-8	2-8	2-8	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8
9_{32}^*	2-8	2-8	1-8	2-8	2-8	1-8	1-8	1-8	2-8	1-8	2-8	1-8	1-8	2-8	1-8

REFERENCES

[1] C.C. Adams. *The Knot Book*. AMS, Providence, Rhode Island, 2004.

[2] A. D. Bates and A. Maxwell. *DNA Topology*. Oxford University Press, Oxford, 2005.

[3] F. Bonahon. *Involutions et fibrés de Seifert dans les variétés de dimension 3*. Thése de 3e cycle, Orsay, 1979.

[4] D. Buck and E. Flapan. A topological characterization of knots and links arising from site-specific recombination. *J. Phys. A*, 40:12377–12395, 2007.

[5] D. Buck and C. Verjovsky Marcotte. Tangle solutions for a family of DNA rearranging proteins. *Math. Proc. Camb. Phil. Soc.*

[6] G. Burde and H. Zieschang. *Knots*. Walter de Gruyter, De Gruyter Studies in Math. 5, 1985.

[7] Hugo Cabrera-Ibarra. On the classification of rational 3-tangles. *J. Knot Theory Ramifications*, 12(7):921–946, 2003.

[8] Hugo Cabrera Ibarra and David A. Lizárraga Navarro. An algorithm based on 3-braids to solve tangle equations arising in the action of Gin DNA invertase. *Appl. Math. Comput.*, 216(1):95–106, 2010.

[9] J.H. Conway. An enumeration of knots and links and some of their algebraic properties. *Computational Problems in Abstract Algebra (John Leech, ed.) Pergamon Press, Oxford and New York*, 1969.

[10] I. K. Darcy. *Biological metrics on DNA knots and catenanes*. PhD thesis, Department of Mathematics, Florida State University, 1997.

[11] I. K. Darcy. Solving oriented tangle equations involving 4-plats. *J. Knot Theory Ramifications*, 14:1007–1027, 2005.

[12] I. K. Darcy. Solving unoriented tangle equations involving 4-plats. *J. Knot Theory Ramifications*, 14:993–1005, 2005.

[13] I. K. Darcy. Modeling protein-DNA complexes with tangles. *Comput. Math. Appl.*, 55(5):924–937, 2008.

[14] I. K. Darcy and R. G. Scharein. Topoice-R: 3d visualization modeling the topology of DNA recombination. *Bioinformatics*, 22(14):1790–1, 2006.

[15] I. K. Darcy, R. G. Scharein, and A. Stasiak. Topoice-X: 3d visualization software to analyze topological outcomes of topoisomerase reactions. *Nucleic Acids Research*, 36(11):3515–3521, 2008.

[16] I. K. Darcy and D. W. Sumners. A strand passage metric for topoisomerase action. In *KNOTS '96 (Tokyo)*, pages 267–278. World Sci. Publishing, River Edge, NJ, 1997.

[17] I. K. Darcy and D. W. Sumners. Applications of topology to DNA. In *Knot theory (Warsaw, 1995)*, volume 42 of *Banach Center Publ.*, pages 65–75. Polish Acad. Sci., Warsaw, 1998.

[18] I. K. Darcy and D. W. Sumners. Rational tangle distances on knots and links. *Math. Proc. Camb. Phil. Soc.*, 128(3):497–510, 2000.

[19] C. Ernst. Tangle equations. *J. Knot Theory Ramifications*, 5(2):145–159, 1996.

[20] C. Ernst and D. W. Sumners. A calculus for rational tangles: applications to DNA recombination. *Math. Proc. Cambridge Philos. Soc.*, 108(3):489–515, 1990.

[21] B. Ewing and K. C. Millett. Computational algorithms and the complexity of link polynomials. In *Progress in knot theory and related topics*, volume 56 of *Travaux en Cours*, pages 51–68. Hermann, Paris, 1997.

[22] T. Fiedler and A. Stoimenow. New knot and link invariants. 1999.

[23] A. Flammini, A. Maritan, and A. Stasiak. Simulations of action of DNA topoisomerases to investigate boundaries and shapes of spaces of knots. *Biophysical Journal*, 87(5):2968–2975, 2004.

[24] P. Freyd, D. Yetter, J. Hoste, W.B.R. Lickorish, K. Millett, and A. Ocneanu. A new polynomial invariant of knots and links. *Bulletin of the American Mathematical Society*, 12(2):239–246, 1985.

[25] C.A. Giller. A family of links and the conway calculus. *Trans. Amer. Math. Soc.*, 270:75–109, 1982.

[26] C. M. Gordon. Dehn surgery and 3-manifolds. *IAS/Park City Mathematics Series*, 16, 2006.

[27] C. McA. Gordon and John Luecke. Knots with unknotting number 1 and essential Conway spheres. *Algebr. Geom. Topol.*, 6:2051–2116 (electronic), 2006.

[28] F. Guo, D. N. Gopaul, and G. D. Van Duyne. Structure of cre recombinase complexed with DNA in a site-specific recombination synapse. *Nature*, 389:40–46, 1997.

[29] J. Hoste and M. Thistlethwaite. Knotscape 1.01.

[30] M. Jankins and W. D. Neumann. Lectures on seifert manifolds. *Brandeis Lecture Notes*, 1983.

[31] V.F.R Jones. A polynomial invariant of knots and links via von neumann algebras. *Bull. Amer. Math. Soc.*, 12:103–111, 1985.

[32] V.F.R Jones. Hecke algebra representation of braid groups and link polynomials. *Ann. of Math.*, 126:335–388, 1987.

[33] L.H. Kauffman and S. Lambropoulou. On the classification of rational tangles. *Advances in Applied Mathematics*, 33:199–237, 2004.

[34] Louis H. Kauffman and Sofia Lambropoulou. Classifying and applying rational knots and rational tangles. In *Physical knots: knotting, linking, and folding geometric objects in* $\mathbb{R}^3$ *(Las Vegas, NV, 2001)*, volume 304 of *Contemp. Math.*, pages 223–259. Amer. Math. Soc., Providence, RI, 2002.

[35] W.B.R. Lickorish and K.C. Millett. Some evaluations of link polynomials. *Comment. Math. Helv*, 61:349–359, 1986.

[36] José M. Montesinos. Surgery on links and double branched covers of S^3. In *Knots, groups, and* 3*-manifolds (Papers dedicated to the memory of R. H. Fox)*, pages 227–259. Ann. of Math. Studies, No. 84. Princeton Univ. Press, Princeton, N.J., 1975.

[37] Hitoshi Murakami. Some metrics on classical knots. *Math. Ann.*, 270(1):35–45, 1985.

[38] K. Murasugi. On a certain numerical invariant of link types. *Trans. Amer. Math. Soc.*, 117:387–422, 1965.

[39] Brendan Owens. Unknotting information from Heegaard Floer homology. *Adv. Math.*, 217(5):2353–2376, 2008.

[40] Peter Ozsváth and Zoltán Szabó. Knots with unknotting number one and Heegaard Floer homology. *Topology*, 44(4):705–745, 2005.

[41] Józef H. Przytycki and Paweł Traczyk. Conway algebras and skein equivalence of links. *Proc. Amer. Math. Soc.*, 100(4):744–748, 1987.

[42] J. Roberts. *Elementary number theory, a problem oriented approach.* MIT Press, Cambridge, MA, 1977.

[43] D. Rolfsen. *Knots and Links.* Publish or Perish, Inc., 1976.

[44] Y. Saka and M. Vazquez. Tanglesolve: topological analysis of site-specific recombination. *Bioinformatics*, 18:1011–1012, 2002.

[45] M. Scharlemann. Unknotting number one knots are prime. *Invent. Math.*, 82:37–55, 1985.

[46] H. Schubert. Knoten mit zwei braucken. *Math. Zeitschrift*, 65:133–170, 1956.

[47] H. Seifert. Seifert and threlfall: a textbook of topology. *Academic Press, Inc., New York - London*, 1980.

[48] A. Stoimenow. Polynomial values, the linking form and unknotting numbers. *Math. Research Letters*, 11:755–768, 2004.

[49] I. Torisu. The determination of the pairs of two-bridge knots or links with Gordian distance one. *Proc. Amer. Math. Soc.*, 5:1565–1571, 1998.

[50] P. Traczyk. A criterion for signed unknotting number. *Contemp. Math.*, 233:215–220, 1999.

[51] M. Vazquez and D. W. Sumners. Tangle analysis of gin site-specific recombination. *Math. Proc. Camb. Phil. Soc.*, 136(3):565–582, 2004.

[52] A. A. Vetcher, A. Y. Lushnikov, J. Navarra-Madsen, R. G. Scharein, Y. L. Lyubchenko, I. K. Darcy, and S. D. Levene. DNA topology and geometry in flp and cre recombination. *Journal of Molecular Biology*, 357(4):1089–104, 2006.

[53] James C. Wang. Cellular roles of dna topoisomerases: a molecular perspective. *Nat Rev Mol Cell Biol*, 3:430–440, June 2002.

[54] X. Zhang. Unknotting number one knots are prime:a new proof. *Proc. AMS*, 113:611–612, 1991.

[55] X.D. Zhu and P.D. Sadowski. Cleavage-dependent ligation by the flp recombinase. *J. Biol. Chem*, 270:23044–23054, 1995.

[56] H. Zieschang. Classification of montesinos knots. *Lect. Notes Math.*, 1060:378–389, 1984.