

Fast Algorithms for Computing Jones Polynomials of Certain Links

Masahiko Murakami[†]
村上 雅彦

Masao Hara[‡]
原 正雄

Makoto Yamamoto[§]
山本 慎

Seiichi Tani[¶]
谷 聖一

Abstract

We give fast algorithms for computing Jones polynomials of 2-bridge links, closed 3-braid links and Montesinos links from a progressive expression. The algorithms run with $\mathcal{O}(n)$ operations of polynomials of degree $\mathcal{O}(n)$, where n is the number of the crossings of the link diagram. We also give linear time algorithms for computing a progressive expression from the Tait graph of a link diagram of 2-bridge links and closed 3-braid links.

1 Introduction

In knot theory, various invariants are defined and studied for classifying and characterizing links. The Jones polynomial [4] is one of the well-studied invariants. It is powerful for distinguishing link types. The simplest way to define it is by using a slightly different polynomial: the bracket polynomial discovered by L.H. Kauffman [5]. But, it takes $\mathcal{O}(2^{\mathcal{O}(\sqrt{c(\tilde{L})})})$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$ to compute a Jones polynomial in the way shown by Kauffman. Actually, computing the Jones polynomial is generally $\#P$ -hard [3, 12] and is expected to require exponential time in the worst case. K. Sekine, H. Imai and K. Imai [10] showed an algorithm that computes Jones polynomials in $\mathcal{O}(2^{\mathcal{O}(\sqrt{c(\tilde{L})})})$ time.

Recently, it has been recognized that it is important to compute Jones polynomials for links with reasonable restrictions. For any link diagram $\tilde{L}$, we denote the number of the crossings of $\tilde{L}$ by $c(\tilde{L})$. J. A. Makowsky [6] showed that Jones polynomials are computed from the Tait graph G of $\tilde{L}$ in polynomial time if the treewidth of G is a constant. J. Mighton [7] showed that Jones polynomials are computed from the Tait graph G of $\tilde{L}$ with $\mathcal{O}(c(\tilde{L})^4)$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$ if the treewidth of G is at most 2. M. Hara, S. Tani and M. Yamamoto [2] showed that Jones polynomials of 2-bridge links are computed from the Tait graph of $\tilde{L}$ with $\mathcal{O}(c(\tilde{L})^2)$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$, and Jones polynomials of closed 3-braid links and arborescent links are computed from the Tait graph of $\tilde{L}$ with $\mathcal{O}(c(\tilde{L})^3)$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$. T. Utsumi and K. Imai [11] showed that Jones polynomials of pretzel links are computed from the Tait graph of $\tilde{L}$ in $\mathcal{O}(c(\tilde{L})^2)$ time.

In this paper, we give algorithms that compute Jones polynomials of 2-bridge links and closed 3-braid links from the Tait graph of $\tilde{L}$ with $\mathcal{O}(c(\tilde{L}))$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$. We also show that Jones polynomials of Montesinos links are computed from a progressive expression of $\tilde{L}$ with $\mathcal{O}(c(\tilde{L}))$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$.

2 Preliminaries

A *link* of n components is a subset of $\mathbb{R}^3$ that consists of n disjoint, simple closed curves. A link of one component is a *knot*. An image of a link by a natural projection from $\mathbb{R}^3$ to a plane is *regular* if it contains only finitely many multiple points, all multiple points are double points and these are traverse points. A regular image of a link is called a *link diagram* if the overcrossing line is marked at every double point in the image. Furthermore, the double points are called *crossings*. For any link diagram $\tilde{L}$, we denote the number of the crossings of $\tilde{L}$ by $c(\tilde{L})$. A link is *oriented* if each of its components is given an orientation.

Definition 2.1 The *Kauffman bracket polynomial* is a function from link diagrams to the Laurent polynomial ring $\mathbb{Z}[A^{\pm 1}]$ with integer coefficients in an indeterminate A . It maps a link diagram $\tilde{L}$ to $\langle \tilde{L} \rangle \in \mathbb{Z}[A^{\pm 1}]$ and is characterized by

[†]Graduate School of Integrated Basic Sciences, Nihon University. 日本大学大学院総合基礎科学研究科

masahiko@tani.cs.chs.nihon-u.ac.jp

[‡]Department of Mathematical Sciences, Tokai University. 東海大学理学部情報数理学科 masao@ss.u-tokai.ac.jp

[§]Department of Mathematics, Chuo University. 中央大学理工学部数学科 makotoy@math.chuo-u.ac.jp

Partially supported by Grant-in-Aid for Science Research No.14540136(C)(2), Japan Society for Promotion of Science and Research Projects for Promotion of Advanced Research at Graduate School, Chuo University.

[¶]Department of Computer Science and System Analysis, Nihon University. 日本大学文理学部情報システム解析学科

sei-ichi@tani.cs.chs.nihon-u.ac.jp

Partially supported by Grant-in-Aid for Science Research No.14580391(C)(2), Japan Society for Promotion of Science.

$$(i) \langle \bigcirc \rangle = 1, \quad (ii) \langle \tilde{L} \sqcup \bigcirc \rangle = (-A^{-2} - A^2) \langle \tilde{L} \rangle, \quad (iii) \langle \times \rangle = A \langle \rangle + A^{-1} \langle \rangle.$$

Here, $\bigcirc$ is the link diagram of the unknot without a crossing and $\tilde{L} \sqcup \bigcirc$ is a link diagram consisting of the link diagram $\tilde{L}$ together with an extra closed curve $\bigcirc$ that contains no crossing at all, neither with itself nor $\tilde{L}$. In (iii) the formula refers to three link diagrams that are exactly the same except near a point where they differ in the way indicated.

The *writhe* $w(\tilde{L})$ of an oriented link diagram $\tilde{L}$ is the sum of the signs of the crossings of $\tilde{L}$, where each crossing has sign +1 or -1 as defined (by convention) in Figure 1.

Definition 2.2 The *Jones polynomial* $V(L)$ of an oriented link L is the Laurent polynomial in $t^{1/2}$ with integer coefficients, defined by

$$V(L) = (-A)^{-3w(\tilde{L})} \langle \tilde{L} \rangle \Big|_{t^{1/2}=A^{-2}},$$

where $\tilde{L}$ is any oriented link diagram for L .

Given any link diagram $\tilde{L}$, we can color the faces black and white in such a way that no two faces with a common edge are the same color. We color the unique unbounded face white. We call this the *Tait coloring* of $\tilde{L}$. As in Figure 2, we can get a signed planar graph G of $\tilde{L}$; its vertices are the black faces of the Tait coloring and two vertices are joined by a signed edge if they share a crossing. The sign of the edge is +1 or -1 according to the (conventional) rule shown in Figure 3. G is the *Tait graph* of $\tilde{L}$.

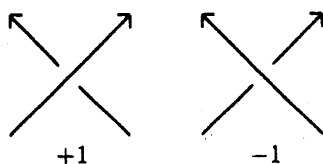


Figure 1: Signs of crossings

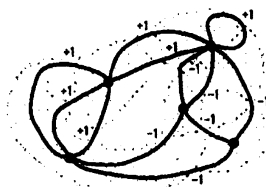


Figure 2: A Tait graph

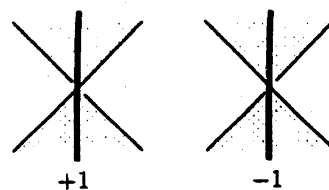


Figure 3: Signs of edges

A *tangle* is a portion of a link diagram from which there emerge just 4 arcs. The tangle consisting of two vertical strings without a crossing is called *0-tangle*. The tangle twisted 0-tangle k times is called *k-tangle* and is denoted by I_k . They are called *integer tangles* (Figure 4). The tangle consisting of two horizontal strings without a crossing is called ∞ -tangle. For a set S , we denote the number of the elements of S by $|S|$. For an integer z , $\text{sign}(z)$ is defined by the following:

$$\text{sign}(z) = \begin{cases} 1 & \text{if } z \geq 0, \\ -1 & \text{if } z < 0. \end{cases}$$

Let $G = (V, E)$ be a graph, where V is the vertex set of G and E is the edge set of G . For any vertex $v \in V$, $\deg_G(v)$ denotes the degree of v in G and $N_G(v)$ denotes the set of the neighbors of v in G . For any subset V' of V , $G[V']$ denotes the induced subgraph of G by V' . For any vertices $u, v \in V$, edge_sign_G is a function from $V \times V$ to $\mathbb{Z}$ and $\text{edge_sign}_G(u, v)$ is the sum of the signs of the edges of G that join u and v .

3 Algorithms for 2-bridge links

Schubert [9] defined a numerical link invariant called bridge number and many knot theorists have been studying it (see [8, 1]). In particular, the 2-bridge link is one of the most important link types. It is well known that any 2-bridge link has a diagram consisting of integer tangles I_{a_k} as in Figure 5 where a_k is an integer for $k = 1, \dots, m$.

Definition 3.1 A link diagram as in Figure 5 is called a *normal diagram* of a 2-bridge link and is denoted by $\tilde{R}(a_1, \dots, a_m)$. The progression $(a_1, \dots, a_m)$ is called a *progressive expression* of the normal diagram.

Lemma 3.2 For any Tait graph $G = (V, E)$ of a normal diagram of a 2-bridge link, there exists a vertex $v \in V$ such that $G[V - \{v\}]$ is a path.

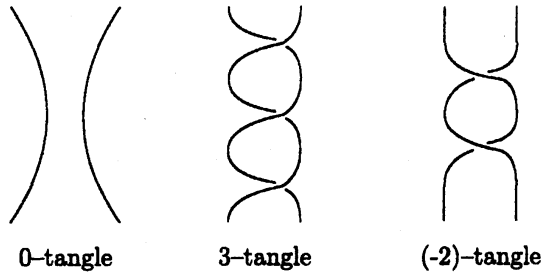
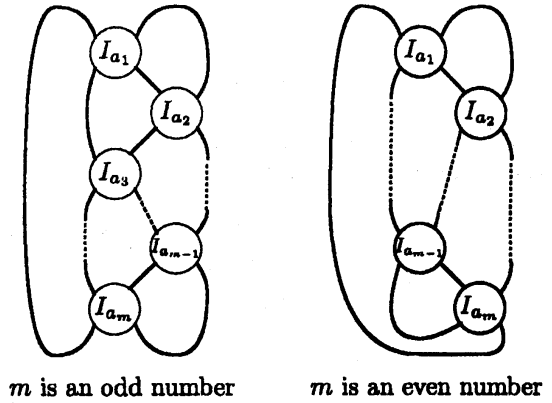


Figure 4: Integer tangles



m is an odd number m is an even number

Figure 5: Normal diagrams of 2-bridge links

Lemma 3.3 For any Tait graph $G = (V, E)$ of a normal diagram of a 2-bridge link, for a vertex $v \in V$, if $\deg_G(v)$ is the maximum degree of G , then $G[V - \{v\}]$ is a path.

Given the Tait graph of a normal diagram $\tilde{L}$ of a 2-bridge link, **Procedure progression_2-bridge** computes a progressive expression of $\tilde{L}$.

Procedure progression_2-bridge

INPUT: The Tait graph $G = (V, E)$ of a normal diagram $\tilde{L}$ of a 2-bridge link.

OUTPUT: A progressive expression $(a_1, \dots, a_m)$ of $\tilde{L}$.

$p \leftarrow |V| - 1$.

Label a vertex $v \in V$ as " v_p " such that $\deg_G(v_p)$ is the maximum degree of G .

Label all vertices $v \in V - \{v_p\}$ as " $v_0, \dots, v_{p-1}$ " such that v_i and v_{i+1} are adjacent for $i = 0, \dots, p-2$.

Compute $\text{edge_sign}_G(v_i, v_p)$ for $i = 0, \dots, p-1$ and $\text{edge_sign}_G(v_j, v_{j+1})$ for $j = 0, \dots, p-2$.

Initialize i as "0" and k as "1".

while $i < p-1$ do

 { k is an odd number }

$a_k \leftarrow -\text{edge_sign}_G(v_i, v_p)$.

 Increment k .

 { k is an even number }

 Initialize a_k as "0".

 repeat

$a_k \leftarrow a_k + \text{edge_sign}_G(v_i, v_{i+1})$.

 Increment i .

 until $\deg_G(v_i) \neq 2$ or $i = p-1$

 Increment k .

od

$a_k \leftarrow -\text{edge_sign}_G(v_{p-1}, v_p)$.

Lemma 3.4 Given the Tait graph of a normal diagram $\tilde{L}$ of a 2-bridge link, **Procedure progression_2-bridge** computes a progressive expression of $\tilde{L}$ in $\mathcal{O}(c(\tilde{L}))$ time.

Lemma 3.5 For any normal diagram $\tilde{R}(a_1, \dots, a_m)$ of a 2-bridge link, the following recurrence formula holds.

$$\langle \tilde{R}(a_1, \dots, a_m) \rangle = \begin{cases} A^{a_1}(-A^{-2} - A^2) + (-A)^{-3a_1 - 2\text{sign}(a_1)} \sum_{k=1}^{|a_1|} (-A^{4\text{sign}(a_1)})^k & \text{if } m = 1, \\ A^{a_2}(-A^{-3})^{a_1} + (-A)^{-3a_2 - 2\text{sign}(a_2)} \langle \tilde{R}(a_1) \rangle \sum_{k=1}^{|a_2|} (-A^{4\text{sign}(a_2)})^k & \text{if } m = 2, \\ A^{a_m}(-A^{-3})^{a_{m-1}} \langle \tilde{R}(a_1, \dots, a_{m-2}) \rangle \\ \quad + (-A)^{-3a_m - 2\text{sign}(a_m)} \langle \tilde{R}(a_1, \dots, a_{m-1}) \rangle \sum_{k=1}^{|a_m|} (-A^{4\text{sign}(a_m)})^k & \text{if } m \geq 3. \end{cases}$$

Given a progressive expression of a normal diagram $\tilde{L}$ of a 2-bridge link, **Procedure bracket_2-bridge** computes the Kauffman bracket polynomial of $\tilde{L}$ by using the recurrence formula in Lemma 3.5. While the algorithm is running, every Kauffman bracket polynomial is computed once at most.

Procedure bracket_2-bridge

INPUT: A progressive expression $(a_1, \dots, a_m)$ of a normal diagram $\tilde{L}$ of a 2-bridge link.

OUTPUT: The Kauffman bracket polynomial $\langle \tilde{R}(a_1, \dots, a_m) \rangle$ of $\tilde{L}$.

Compute $\langle \tilde{R}(a_1) \rangle$ and $\langle \tilde{R}(a_1, a_2) \rangle$.

Initialize i as "3".

while $i \leq m$ do

$$T \leftarrow \sum_{k=1}^{|a_i|} \left(-A^{4\text{sign}(a_i)} \right)^k.$$

Compute $\langle \tilde{R}(a_1, \dots, a_i) \rangle$ from $\langle \tilde{R}(a_1, \dots, a_{i-2}) \rangle$, $\langle \tilde{R}(a_1, \dots, a_{i-1}) \rangle$ and T .

Increment i .

od

Lemma 3.6 Given a progressive expression of a normal diagram $\tilde{L}$ of a 2-bridge link, **Procedure bracket_2-bridge** computes the Kauffman bracket polynomial of $\tilde{L}$ with $\mathcal{O}(c(\tilde{L}))$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$.

Theorem 3.7 The Jones polynomial of a 2-bridge link is computed from the Tait graph of a normal diagram $\tilde{L}$ of the 2-bridge link with $\mathcal{O}(c(\tilde{L}))$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$.

4 Algorithms for closed 3-braid links

A 3-braid is a set of 3 strings, all of which are attached to a horizontal bar at the top and at the bottom and each string intersects any horizontal plane between the two bars exactly once (see Figure 6 (a)). Given any 3-braid, its ends on the bottom edge may be joined to those on the top edge to produce the *closed 3-braid link* (see Figure 6 (b)). It is clear that any closed 3-braid link has a diagram consisting of integer tangles I_{a_k} as in Figure 7 where a_k is an integer for $k = 1, \dots, m$.

Definition 4.1 A link diagram as in Figure 7 is called a *normal diagram* of a closed 3-braid link and is denoted by $\tilde{B}(a_1, \dots, a_m)$. The progression $(a_1, \dots, a_m)$ is called a *progressive expression* of the normal diagram.

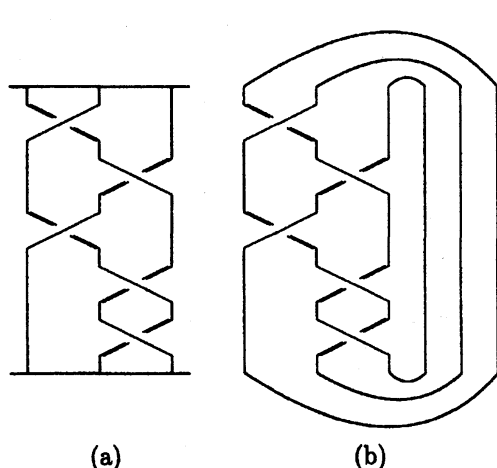
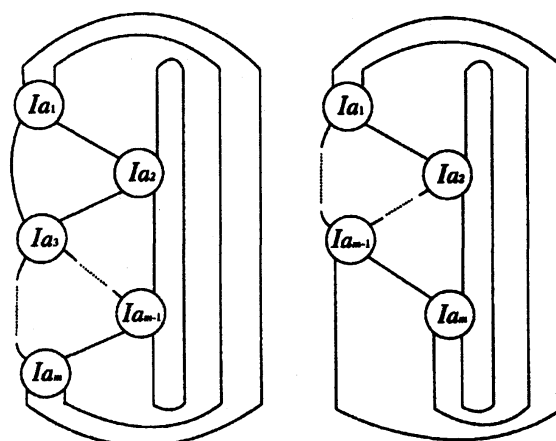


Figure 6: (a) A 3-braid, (b) A closed 3-braid link



m is an odd number

m is an even number

Figure 7: Normal diagrams of closed 3-braid links

Lemma 4.2 For any Tait graph $G = (V, E)$ of a normal diagram of a closed 3-braid link, there exists a vertex $v \in V$ such that $G[V - \{v\}]$ is a cycle.

Given the Tait graph of a normal diagram $\tilde{L}$ of a closed 3-braid link, **Procedure progression_3-braid** computes a progressive expression of $\tilde{L}$.

Procedure progression_3-braid

INPUT: The Tait graph $G = (V, E)$ of a normal diagram $\tilde{L}$ of a closed 3-braid link.

OUTPUT: A progressive expression $(a_1, \dots, a_m)$ of $\tilde{L}$.

$p \leftarrow |V| - 1$.

if there exists a vertex $v \in V$ such that $|N_G(v)| \geq 4$

Label v as " v_p ",

else if there exists a vertex $v \in V$ such that $|N_G(v)| = 0$

Label v as " v_p ",

else if there exists a vertex $v \in V$ such that $|N_G(v)| = 1$

Label v as " v_p ",

else if there exists no vertex $v \in V$ such that $|N_G(v)| \neq 2$ { for any vertex $v \in V$, $|N_G(v)| = 2$ }

Label a vertex $v \in V$ as " v_p " such that $G[V - \{v_p\}]$ is a cycle,

else

$V' \leftarrow \{v \mid v \in V, |N_G(v)| = 3\}$,

Label a vertex $v \in V$ as " v_p " such that $\bigcap_{v' \in V' - \{v_p\}} N_G(v') = \{v_p\}$.

Label the all vertices $v \in V - \{v_p\}$ as " $v_0, \dots, v_{p-1}$ "

such that v_i and $v_{i+1 \bmod p}$ are adjacent for $i = 0, \dots, p-1$.

Compute $\text{edge_sign}_G(v_i, v_p)$ and $\text{edge_sign}_G(v_i, v_{i+1 \bmod p})$ for $i = 0, \dots, p-1$.

Initialize i as "0" and k as "1".

repeat

{ k is an odd number }

$a_k \leftarrow -\text{edge_sign}_G(v_i, v_p)$.

Increment k .

{ k is an even number }

Initialize a_k as "0".

repeat

$a_k \leftarrow a_k + \text{edge_sign}_G(v_i, v_{i+1 \bmod p})$.

Increment i .

until $\deg_G(v_i) \neq 2$ or $i = p$

Increment k .

until $i = p$

Lemma 4.3 Given the Tait graph of a normal diagram $\tilde{L}$ of a closed 3-braid link, **Procedure progression_3-braid** computes a progressive expression of $\tilde{L}$ in $\mathcal{O}(c(\tilde{L}))$ time.

Lemma 4.4 For any normal diagram $\tilde{B}(a_1, \dots, a_m)$ of a closed 3-braid link, the following recurrence formula holds.

$$\langle \tilde{B}(a_1, \dots, a_m) \rangle = \begin{cases} (-A^{-2} - A^2) \langle \tilde{R}(a_1) \rangle & \text{if } m = 1, \\ A^{a_m} \langle \tilde{B}(a_1, \dots, a_{m-1}) \rangle + (-A)^{-3a_m - 2\text{sign}(a_m)} & \text{if } m \geq 2 \text{ and} \\ \quad \times \langle \tilde{R}(a_1, \dots, a_{m-1}) \rangle \sum_{k=1}^{|a_m|} (-A^{4\text{sign}(a_m)})^k & m \text{ is an even number,} \\ A^{a_m} \langle \tilde{B}(a_1, \dots, a_{m-1}) \rangle + (-A)^{-3(a_m + a_1) - 2\text{sign}(a_m)} & \text{if } m \geq 3 \text{ and} \\ \quad \times \langle \tilde{R}(a_2, \dots, a_{m-1}) \rangle \sum_{k=1}^{|a_m|} (-A^{4\text{sign}(a_m)})^k & m \text{ is an odd number.} \end{cases}$$

Given a progressive expression of a normal diagram $\tilde{L}$ of a closed 3-braid link, **Procedure bracket_3-braid** computes the Kauffman bracket polynomial of $\tilde{L}$ by using the recurrence formulas in **Lemma 3.5** and **Lemma 4.4**. While the algorithm is running, every Kauffman bracket polynomial is computed once at

most.

Procedure bracket_3-braid

INPUT: A progressive expression $(a_1, \dots, a_m)$ of a normal diagram $\tilde{L}$ of a closed 3-braid link.

OUTPUT: The Kauffman bracket polynomial $\langle \tilde{B}(a_1, \dots, a_m) \rangle$ of $\tilde{L}$.

Compute $\langle \tilde{R}(a_1) \rangle$, $\langle \tilde{B}(a_1) \rangle$, $\langle \tilde{B}(a_1, a_2) \rangle$, $\langle \tilde{R}(a_1, a_2) \rangle$, $\langle \tilde{R}(a_2) \rangle$,
 $\langle \tilde{B}(a_1, a_2, a_3) \rangle$, $\langle \tilde{R}(a_1, a_2, a_3) \rangle$ and $\langle \tilde{R}(a_2, a_3) \rangle$.

Initialize i as "4".

while $i \leq m$ do

$$T \leftarrow \sum_{k=1}^{|a_i|} \left(-A^{4\text{sign}(a_i)} \right)^k.$$

if i is an even number

Compute $\langle \tilde{B}(a_1, \dots, a_i) \rangle$ from $\langle \tilde{B}(a_1, \dots, a_{i-1}) \rangle$, $\langle \tilde{R}(a_1, \dots, a_{i-1}) \rangle$ and T ,

else

Compute $\langle \tilde{B}(a_1, \dots, a_i) \rangle$ from $\langle \tilde{B}(a_1, \dots, a_{i-1}) \rangle$, $\langle \tilde{R}(a_2, \dots, a_{i-1}) \rangle$ and T .

Compute $\langle \tilde{R}(a_1, \dots, a_i) \rangle$ from $\langle \tilde{R}(a_1, \dots, a_{i-2}) \rangle$, $\langle \tilde{R}(a_1, \dots, a_{i-1}) \rangle$ and T .

Compute $\langle \tilde{R}(a_2, \dots, a_i) \rangle$ from $\langle \tilde{R}(a_2, \dots, a_{i-2}) \rangle$, $\langle \tilde{R}(a_2, \dots, a_{i-1}) \rangle$ and T .

Increment i .

od

Lemma 4.5 Given a progressive expression of a normal diagram $\tilde{L}$ of a closed 3-braid link, **Procedure bracket_3-braid** computes the Kauffman bracket polynomial of $\tilde{L}$ with $\mathcal{O}(c(\tilde{L}))$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$.

Theorem 4.6 The Jones polynomial of a closed 3-bridge link is computed from the Tait graph of a normal diagram $\tilde{L}$ of the closed 3-braid link with $\mathcal{O}(c(\tilde{L}))$ operations of polynomials of degree $\mathcal{O}(c(\tilde{L}))$.

5 Algorithms for Montesinos links

Definition 5.1 A link that has a link diagram as in Figure 8 is a *Montesinos link*. The link diagram is called a *normal diagram* of the Montesinos link and is denoted by $\tilde{M}(a_{11}, \dots, a_{1m_1} | \dots | a_{l1}, \dots, a_{lm_l})$. The progression $(a_{11}, \dots, a_{1m_1} | \dots | a_{l1}, \dots, a_{lm_l})$ is called a *progressive expression* of the normal diagram. A link diagram as in Figure 9 is denoted by $\tilde{N}(a_{11}, \dots, a_{1m_1} | \dots | a_{l1}, \dots, a_{lm_l})$.

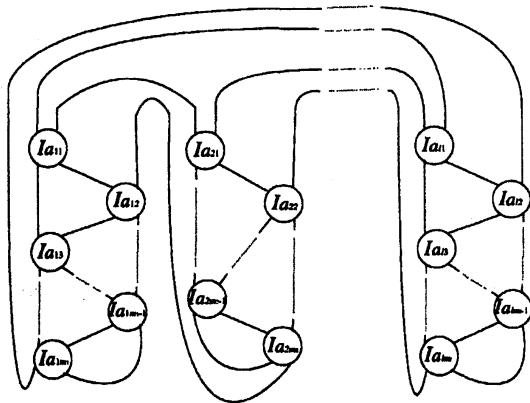


Figure 8: A normal diagram of a Montesinos link

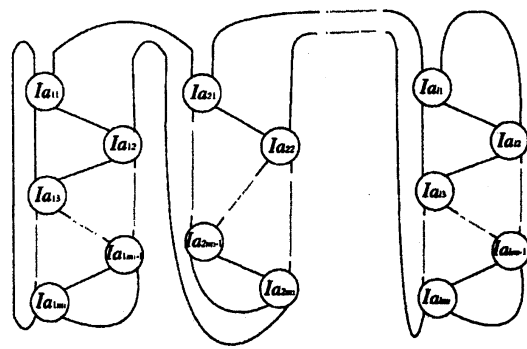


Figure 9: $\tilde{N}(a_{11}, \dots, a_{1m_1} | \dots | a_{l1}, \dots, a_{lm_l})$

Lemma 5.2 For any normal diagram $\tilde{M}(a_{11}, \dots, a_{1m_1} | \dots | a_{l1}, \dots, a_{lm_l})$ of a Montesinos link, the following recurrence formula holds.

$$\begin{aligned}
& \langle \widetilde{M}(a_{11}, \dots, a_{1m_1} | \dots | a_{l1}, \dots, a_{lm_l}) \rangle \\
= & \begin{cases} (-A^{-3})^{a_{11}} & \text{if } l = 1 \text{ and } m_1 = 1, \\ \\ (-A^{-3})^{a_{11}} \langle \widetilde{R}(a_{12}, \dots, a_{1m_1}) \rangle & \text{if } l = 1 \text{ and } m_1 \geq 2, \\ \\ A^{a_{11}} \langle \widetilde{N}(a_{11}, \dots, a_{1m_1} | \dots | a_{l-11}, \dots, a_{l-1m_{l-1}}) \rangle \\ + (-A)^{-3a_{11}-2\text{sign}(a_{11})} \langle \widetilde{M}(a_{11}, \dots, a_{1m_1} | \dots | a_{l-11}, \dots, a_{l-1m_{l-1}}) \rangle \\ \times \sum_{k=1}^{|a_{11}|} (-A^{4\text{sign}(a_{11})})^k & \text{if } l \geq 2 \text{ and } m_l = 1, \\ \\ A^{a_{12}} (-A^{-3})^{a_{11}} \langle \widetilde{M}(a_{11}, \dots, a_{1m_1} | \dots | a_{l-11}, \dots, a_{l-1m_{l-1}}) \rangle \\ + (-A)^{-3a_{12}-2\text{sign}(a_{12})} \langle \widetilde{M}(a_{11}, \dots, a_{1m_1} | \dots | a_{l-11}, \dots, a_{l-1m_{l-1}} | a_{l1}) \rangle \\ \times \sum_{k=1}^{|a_{12}|} (-A^{4\text{sign}(a_{12})})^k & \text{if } l \geq 2 \text{ and } m_l = 2, \\ \\ A^{a_{1m_l}} (-A^{-3})^{a_{1m_l-1}} \langle \widetilde{M}(a_{11}, \dots, a_{1m_1} | \dots | a_{l1}, \dots, a_{lm_l-2}) \rangle \\ + (-A)^{-3a_{1m_l}-2\text{sign}(a_{1m_l})} \langle \widetilde{M}(a_{11}, \dots, a_{1m_1} | \dots | a_{l1}, \dots, a_{lm_l-1}) \rangle \\ \times \sum_{k=1}^{|a_{1m_l}|} (-A^{4\text{sign}(a_{1m_l})})^k & \text{if } l \geq 2 \text{ and } m_l \geq 3. \end{cases}
\end{aligned}$$

For any $\widetilde{N}(a_{11}, \dots, a_{1m_1} | \dots | a_{l1}, \dots, a_{lm_l})$, we also get a recurrence formula.

Theorem 5.3 *The Jones polynomial of a Montesinos link is computed from a progressive expression of a normal diagram $\widetilde{L}$ of the Montesinos link with $\mathcal{O}(c(\widetilde{L}))$ operations of polynomials of degree $\mathcal{O}(c(\widetilde{L}))$.*

References

- [1] G. Burde, H. Zieschang. *Knots*, Walter de Gruyter, 1985.
- [2] M. Hara, S. Tani, M. Yamamoto. A polynomial-time algorithm for computing the Jones polynomials of arborescent links (in Japanese). *Information Technology Letters* Vol. 1, pp.16-17 (2002).
- [3] F. Jaeger, D. L. Vertigan, and D. J. A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Math. Proc. Camb. Phil. Soc.*, 108:35-53, 1990.
- [4] V. F. R. Jones. A polynomial invariant for knots via Von Neumann algebras. *Bull. Amer. Math. Soc.*, 12:103-111, 1985.
- [5] L. H. Kauffman. State models and the Jones polynomial. *Topology*, 26:395-407, 1987.
- [6] J. A. Makowsky. Colored Tutte polynomials and Kauffman brackets for graphs of bounded tree width. Preprint.
- [7] J. Mighton. *Knot Theory on Bipartite Graphs*. PhD thesis, Dept. of Math., University of Toronto, Canada, 1999.
- [8] D. Rolfsen. *Knots and Links*, Publish or Perish, Inc., 1976.
- [9] H. Schubert. Über eine Numerische Knoteninvariante. *Math. Z.*, 61:245-288, 1954.
- [10] K. Sekine, H. Imai, K. Imai. Computation of Jones Polynomial (in Japanese). *Transaction of the Japan Society for Industrial and Applied Mathematics*, 8 No. 3 (1998) 341-354.
- [11] T. Utsumi, K. Imai. Computation of the Jones Polynomials for Pretzel Links (in Japanese). *IPSI SIG Technical Reports* No.085 (2002).
- [12] D. J. A. Welsh. *Complexity: Knots, Colorings and Counting*, Cambridge Univ. Press, 1993.