Università degli studi di Camerino

Scuola di Scienze e Tecnologie

Corso di Laurea in Matematica e applicazioni
(classe LM-40)

QUANTUM COMPUTATION
OF THE JONES POLYNOMIAL

Tesi di Laurea in Topologia

Relatore:
Prof. Riccardo Piergallini

Laureanda:
Alessandra Renieri

Anno Accademico 2010-2011

Abstract

Knot invariants are algebraic objects associated to knots, which do not change under isotopies. Such invariants are important tools for the classification of knots, since they allow to distinguish knots that are not isotopic.

The Jones polynomial is one of the most important knot invariants. The definition of it, given by Vaughan Jones in 1984, is based on the realization of knots as closed braids and on algebraic notions inspired to Quantum Physics, such as Heck algebras and the Yang-Baxter equation. Some years later, Louis Kauffman provided a different and simpler approach to the Jones polynomial, based on a recursive scheme on the number of crossings of a diagram of the knot.

Both these approaches lead to classical algorithms for the computation of the Jones polynomial, which are not efficient, that is their complexity grows exponentially with the number of the crossings of the braid or diagram.

On the contrary, in the context of Quantum Computation the problem admits a solution having polynomial complexity.

The aim of this work is to illustrate an explicit quantum algorithm for approximating the value $V_K(e^{2\pi i/k})$ of the Jones polynomial at any root of unity $t = e^{\frac{2\pi i}{k}}$. More precisely, the algorithm takes as the input a knot K represented by a closed braid with n strands and m crossing, and produces as the output an ε-approximation of the value $V_K(e^{2\pi i/k})$ in a polynomial time with respect to m, n, k and $\frac{1}{\varepsilon}$, with all but exponentially small probability.

Contents

List of Figures

Introduction

Knot invariants are algebraic objects associated to knots, which do not change under isotopes. Such invariants are important tools for the classification of knots, since they allow to distinguish knots that are not isotopic.

The Jones polynomial is one of the most important knot invariants. The definition of it, given by Vaughan Jones in 1984, is based on the realization of knots as closed braids and on algebraic notions inspired to Quantum Physics, such as Heck algebras and the Yang-Baxter equation. Some years later, Louis Kauffman provided a different and simpler approach to the Jones polynomial, based on a recursive scheme on the number of crossings of a diagram of the knot.

Both these approaches lead to classical algorithms for the computation of the Jones polynomial, which are not efficient, that is their complexity grows exponentially with the number of the crossings of the braid or diagram.

On the contrary, in the context of Quantum Computation the problem admits a solution having polynomial complexity.

The aim of this work is to illustrate an explicit quantum algorithm for approximating the value $V_K(e^{2\pi i/k})$ of the Jones polynomial at any root of unity $t = e^{\frac{2\pi i}{k}}$. More precisely, the algorithm takes as the input a knot K represented by a closed braid with n strands and m crossing, and produces as the output an ε-approximation of the value $V_K(e^{2\pi i/k})$ in a polynomial time with respect to m, n, k and $\frac{1}{\varepsilon}$, with all but exponentially small probability.

The thesis is divided in five chapters.

In Chapter 1 we introduce the basic notion of Knot theory (see [1],[2],[3]). This theory studies the topology of closed curves in the space, up to isotopy. We also give the definitions of some of the most important knot invariants, such as the Kauffman and the Jones polynomial.

In Chapter 2 we treat the notion of braid (see [5],[9]). Starting from the geometric braids, we have defined the braid group B_n. In order to establish the connecting between braids and knots ([11]), we define two different closures of a braid (trace and plat closure). According to the Alexander theorem we know that for every knot K there exist some braid b in B_n such that the closure of b is K. Then, we give an algebraic description the Kauffman and Jones polynomials based on the notion of Heck algebra and the definition of the function Tr_n, which lead Jones himself to the definition of his polynomial.

In Chapter 3 we define the Temperley Lieb algebra $TL_n(d)$ and, through the construction of a function ρ_d, we represent the group B_n inside $TL_n(d)$. Since the algorithm that we are going to describe is quantistic, the representation that we have is required to be unitary, so that a quantum computer can executes it. Indeed a unitary interpretation of the above representation is given in terms of the so called *path model*. Then we review the Kauffman and the Jones polynomial in this context.

The Chapter 4 is a compendium of the most important notion of the Quantum Computation ([6]), as the four principles of the Quantum Mechanics, the quantum Turing machine, some of the quantum classes of computational complexity and some of the quantum logic gates (as, for example the Hadamard gate).

Chapter 5 is the core of our thesis. Here we describe the approximation algorithm presented in ([8]) and try to compare this quantum algorithm with the classical algorithms for the computation of the exact Jones polynomial.

Chapter 1

Knots

This first chapter is a presentation of the most basic and important objects of the theory of knots. Precisely, we define the notion of knot (and link), the equivalence between two knots, the way a knot can be drawn in a plane, the Reidemeister moves for such planar representations, and some of the invariants that can be associated to a knot. In particular, we describe the Kauffman and the Jones polynomials.

1.1 Knots and diagrams

The mathematical notion of knot arises as an abstract model of the physical object consisting of a loop of rope arbitrarily entangled in the space. This can be continuously deformed, with the obvious physical constraint that it cannot cross itself, but it cannot be cut. Hence, it is natural to expect that the mathematical concept of knot is a topological one.

Definition 1.1.1. *A knot is a closed and simple curve $K \subset \mathbb{R}^3$ such that $K \cong S^1$ (K is topologically equivalent to the circumference). Furthermore, we assume that K is smooth, i.e. the curve can be approximated by its tangent line at any point.*

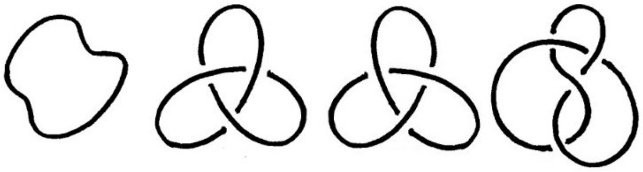


Figure 1.1: The unknot, the right handed trefoil knot and the left handed trefoil knot and the figure eight knot.

The topological union of n knots is said to be a n-link L, i.e.

$$L = K_1 \sqcup \cdots \sqcup K_n \subset \mathbb{R}^3 \ ,$$

where $K_1, \ldots, K_n \subset R^3$ are called the components of L.

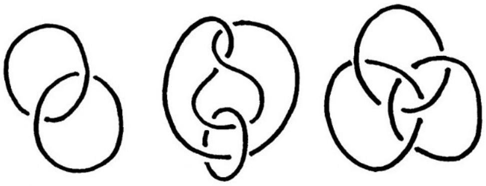


Figure 1.2: Examples of links.

According to the definition a knot K is a special case of a link L with only one component. Henceforth, we will use K to denote both a link or a knot, and we will specify the number of the components only when we will need it.

Sometimes it is convenient to fix an orientation on a knot, meaning one of the two possible ways of running along it. Similarly, one can orient an n-link, by fixing an orientation on each component of it. Of course, this can be done in 2^n ways.

Definition 1.1.2. *An oriented knot is a knot with a specified orientation*

Similarly, we define an oriented link as a link with a specified orientation on each component.

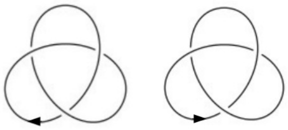


Figure 1.3: The two possible orientations of the left-handed trefoil knot.

Definition 1.1.3. *We define a symmetric knot as a knot equivalent to its mirror image.*

The mathematical formalization of the intuitive idea of deformation is provided by the notion of isotopy introduced in the next definition. For technical reasons, this involves all the space and not only the knot, as one could expect.

Definition 1.1.4. *We define an isotopy of the space $\mathbb{R}^3$ as a continuous application $H : \mathbb{R} \times [0,1] \to \mathbb{R}$, such that the map $h_t : \mathbb{R} \to \mathbb{R}$ given by $x \mapsto h_t(x) = H(x,t)$ is a homomorphism for every $t \in [0,1]$ and moreover h_0 is the identity of $\mathbb{R}^3$*

Then, the concept of isotopy, allows us to define the equivalence between two knots (or links), as follows.

Definition 1.1.5. *Two links $K_0, K_1 \subset \mathbb{R}^3$ are said to be equivalent (or isotopic) if there exists an isotopy of $\mathbb{R}^3$ that transforms K_0 in K_1, i.e. $\exists h_t : \mathbb{R}^3 \to \mathbb{R}^3$ with $t \in [0,1] : h_0 =$identity and $h_1(K_0) = K_1$.*

We observe that the intuitive idea of deformation corresponds to following the knot during the isotopy, that is to considering the continuous family of knots $K_t = h_t(K_0)$. In fact, in Definition 1.1.1 we assumed that knots are smooth and any continuous smooth family of smooth knots extend to an isotopy of the ambient space $\mathbb{R}^3$.

Moreover, it is easy to verify that the equivalence of knots is a genuine relation of equivalence.

In order to draw a knot (or a link) in a plane, so that the picture gives a full representation of it up to isotopy, we can proceed as described below.

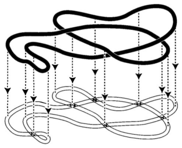


Figure 1.4: Projection of a knot.

We first isotope the original knot (or link) to a suitable one, whose orthogonal projection in $\mathbb{R}^2$ satisfy the following properties:

1. the projection is a regular map, i.e there is not vertical tangency;
2. no more than two distinct points are projected in the same point;
3. the set of the *crossing points*, that is the double points projection, is finite and the projection of the corresponding tangents are distinct.

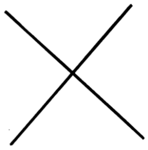


Figure 1.5: Detail: a double point in a projection.

Definition 1.1.6. *A diagram $D \subset \mathbb{R}^2$ of a link L is a projection with specification of what path passes over and which under at each crossing point.*

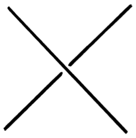


Figure 1.6: Detail: a double point in a diagram.

The representation of link K by a diagram $D \in \mathbb{R}^2$ determines univocally the link K, up to vertical isotopies. We can ask when two diagrams represent equivalent knots.

The answer to this question, was given by Reidemeister [3], in terms of *plane isotopy*, preserving the structure of the diagram, included the information relative to the crossings, and the following *three moves*, which change the topological structure of the diagram.

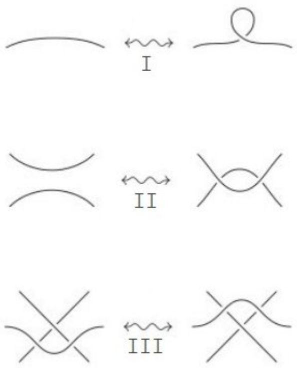


Figure 1.7: The three Reidemeister moves.

Theorem 1.1.7. *Two diagrams D_0 and D_1 represent isotopic links K_0, K_1 if and only if they can be obtained one from the other by a finite sequence of planar isotopies and Reidemeister moves.*

(For the proof see the first Chapter of [4])

This means that the Reidemeister moves reduce the three-dimensional (and abstract problem) of *knot equivalence* to another problem that is two-dimensional (and more concrete).

1.2 Some numerical invariants

In order to classify links we assign to them *invariants*. These are algebraic objects (polynomials or simply numbers) that do not change under topological deformation, that is they only depend on the isotopy class of the link. All the invariants we will consider are defined starting from a diagram of the link. Hence, we have to check that they are invariant under the Reidemeister moves, that is do not depend on the specific diagram of the link.

The crossing number $c(K)$ of a link K is the minimum number of crossings of any diagram D, considered up to isotopy. Being the minimum among all the diagrams of the link K, it does not depend on a specific diagram. Thus no check of invariance with respect to the Reidemeister moves is needed.

For example, using symmetric knots, we know that the unknot has crossing number 0, while the trefoil knots and the figure-eight knot have crossing number respectively 3 and 4. There are no other knots with a crossing number less than 5, and just two knots have crossing number 5. The number of knots with a particular crossing number rapidly increases as the crossing number increases: as it can be seen in the figure below we have three knots with crossing number 6, six knots with crossing number 7, and twenty-one knots with crossing number 8.

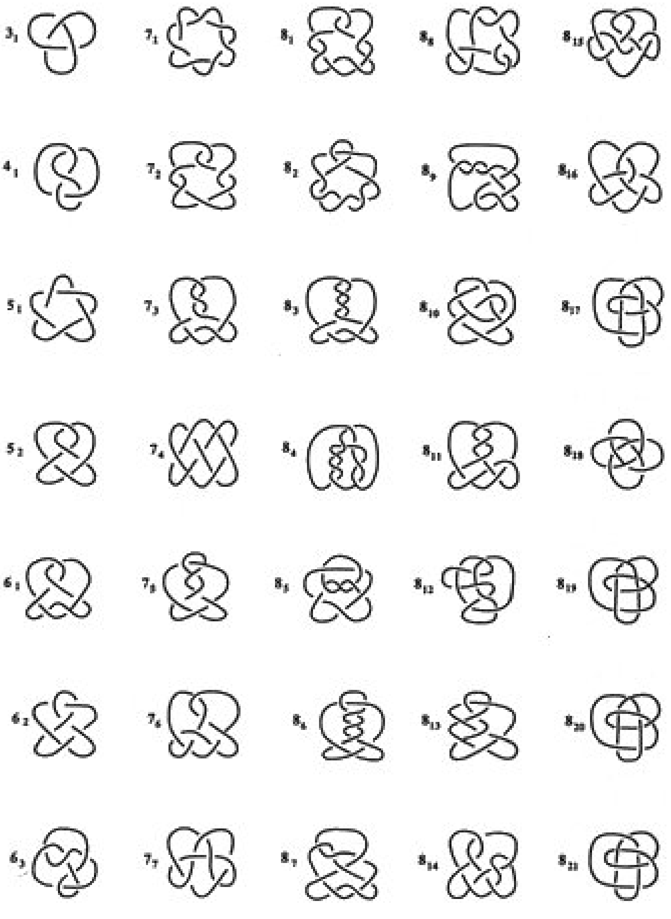


Figure 1.8: The knot K with $c(K) \leq 8$ up to isotopy and symmetry.

The linking number $lk(K)$ of a 2-component link K is the number of times that each curve winds around the other. It can be positive or negative depending on the orientation of the two curves. In fact, every crossing in a diagram of an oriented link has a sign. The crossing can be positive $(+1)$ if the oriented arc passing over has to be rotated through $+90$ degrees in order to fit with that passing under and is negative (-1) if the arc that moves over have to rotate through -90 degrees in order to fit the arc that moves under, as the standard convention proves:

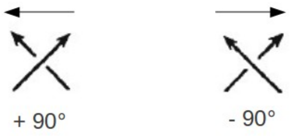
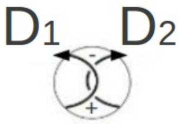


Figure 1.9: The standard sign convection.

Definition 1.2.1. *Let $K = K_1 \sqcup K_2$ be an oriented link and let $D = D_1 \sqcup D_2$ the diagram of it. Then, we have that $lk(K_1, K_2)_D$ is the number of the crossings at which D_1 passes over (or under) D_2, considered with the sign given by the convection above. Equivalently the linking number $lk(K1, K2)_D$ is half the sum of the signs of the crossings at which one strand is from K_1 and the other is from K_2.*

We can see that this number is invariant for Reidemeister moves:

1. the first move involves only one component, thus it does not change the linking number;
2. in the second move, if both the crossings belongs to K_1 or K_2 the linking number does not change as in previous case, otherwise

   the two crossings have opposite signs; thus, also in that case the linking number does not change;

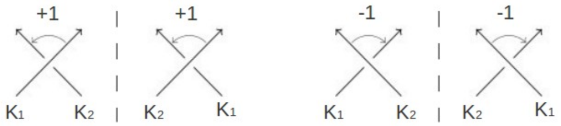
3. this move does not affect the crossing points but only their position in the diagram.

Proposition 1.2.2. *1. $lk(K_1, K_2) = lk(K_2, K_1)$;*

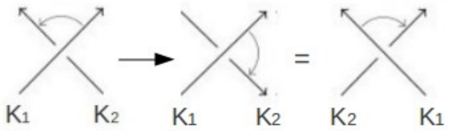
2. $lk(K_1, -K_2) = -lk(K_1, K_2) = lk(-K_1, K_2)$.

Proof. 1. The linking number is symmetric. While $lk(K_1, K_2)$ computes the crossings where K_1 passes over K_2, $lk(K_2, K_1)$ computes the crossings where K_2 passes over K_1.

We take the knot $K = K_1 \sqcup K_2$. Then, rotating the knot, the component K_1 passes under K_2 and thus $lk(K_1, K_2) = lk(K_2, K_1)$ as it is shown in the figure:



2. $lk(-K_1, K_2) = -lk(K_1, K_2)$. Inverting the orientation of a component, all the crossings with the other component change their sign.



□

1.3 The Kauffman-Jones polynomial

This invariant is not a genuine polynomial, is instead a Laurent polynomial, i.e. a linear combination of positive and negative powers of the variable with coefficients in $\mathbb{K}$ field. These Laurent polynomials in x form a ring denoted with $\mathbb{K}[x, x^{-1}]$.

In order to define this invariant we need the concept of state of a diagram.

A state S of a diagram D is a set of plane curves, obtained from the diagram D by replacing each crossing by two segments that do not cross in one of the two ways shown in the following figure:

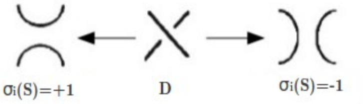


Figure 1.10: The two possible resolutions of a crossing.

For every state we consider:

$\sigma(S) \ = \sum_{i=1}^{n} \sigma_i(S)$ with $\sigma_i(S) = \pm 1$ resolution index of each crossing;

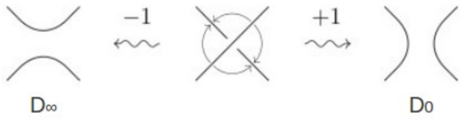
$\gamma(S) \ =$ number of components of the state.

Definition 1.3.1. *The Kauffman brackets of D is the Laurent polynomial in $\mathbb{Z}[x, x^{-1}]$ given by the sum over all the 2^n possible state :*

$$\langle D \rangle = \sum_{S} x^{\sigma(S)} (-x^2 - x^{-2})^{\gamma(S)-1} \ .$$

Property 1.3.2. *The Kauffman brackets are characterized by the following properties:*

1. $\langle \bigcirc \rangle = 1$*;*
2. $\langle \widehat{D} \rangle = (-x^2 - x^{-2}) \langle D \rangle$ *with* $\widehat{D} = D \sqcup \bigcirc$*;*
3. $\langle D \rangle = x \langle D_0 \rangle + x^{-1} \langle D_\infty \rangle$ *where* D_0 *and* D_∞ *are the two possible diagrams that we obtain after the elimination of the same crossing of D in the two possible ways:*



Proof. 1. Trivial proof.

2. $\widehat{D}$ has the same number of crossing of D. For any state S of D we add $\bigcirc$, then the corresponding state $\widehat{S}$ of $\widehat{D}$ is $S \sqcup \bigcirc$ and $\gamma(\widehat{S}) = \gamma(S) + 1$. Thus,

$$\langle \widehat{D} \rangle = \sum_{\widehat{S}} x^{\sigma(\widehat{S})}(-x^2 - x^{-2})^{\gamma(\widehat{S})-1} =$$

$$= \sum_{S} x^{\sigma(S)}(-x^2 - x^{-2})^{\gamma(S)-1}(-x^2 - x^{-2}) = (-x^2 - x^{-2})\langle D \rangle \ .$$

3. We observe that the set of the states S of D are the disjoint union of the set of the states of D_0, that we called S_0, and the set of the states of D_∞, that we called S_∞.

   Since $\sigma(S) = \sigma(S_0) + 1 = \sigma(S_\infty) - 1$ and $\gamma(S) = \gamma(S_0) = \gamma(S_\infty)$,

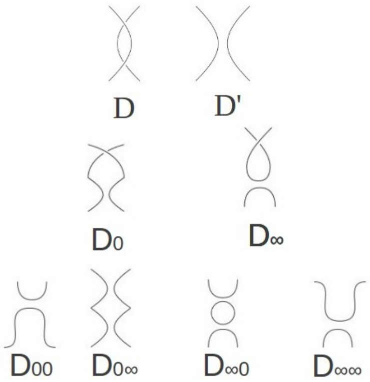
$$\langle D \rangle = \sum_{S_0} x^{\sigma(S_0)+1}(-x^2 - x^{-2})^{\gamma(S_0)-1} + \sum_{S_\infty} x^{\sigma(S_\infty)-1}(-x^2 - x^{-2})^{\gamma(S_\infty)-1} =$$

$$= x\langle D_0 \rangle + x^{-1}\langle D_\infty \rangle$$

□

Property 1.3.3. *The Kauffman brackets are invariant with respect to the II and the III Reidemeister moves.*

Proof. Second move:



since

$$D_{00} \simeq D_{\infty\infty} \ ,$$

$$D_{\infty 0} = D_{\infty\infty} \sqcup \bigcirc = D_{00} \sqcup \bigcirc = \widehat{D}_{00} \ ,$$

$$D_{0\infty} = D',$$

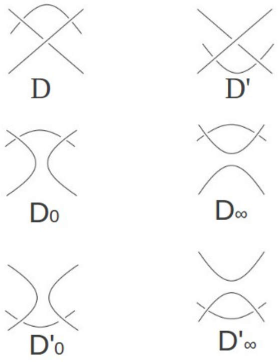
then

$$\langle D \rangle = x\langle D_0 \rangle + x^{-1}\langle D_\infty \rangle = x(x\langle D_{00} \rangle + x^{-1}\langle D_{0\infty} \rangle) + x^{-1}(x\langle D_{\infty 0} \rangle + x^{-1}\langle D_{\infty\infty} \rangle) =$$

$$= x^2\langle D_{00} \rangle + \langle D' \rangle + \langle \widehat{D}_{00} \rangle + x^{-2}\langle D_{00} \rangle = x^2\langle D_{00} \rangle + x^{-2}\langle D_{00} \rangle + \langle D' \rangle + (-x^2 - x^{-2})\langle D_{00} \rangle =$$

$$= (-x^2 - x^{-2})\langle D_{00} \rangle + (x^2 + x^{-2})\langle D_{00} \rangle + \langle D' \rangle = \langle D' \rangle$$

Third move:

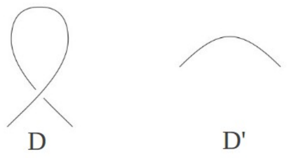


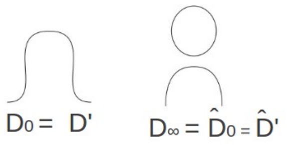
since $D_0 \simeq D'_0$ and $D_\infty = D'_\infty$ for the second Reidemeister move, then

$$\langle D\rangle = x\langle D_0\rangle + x^{-1}\langle D_\infty\rangle = x\langle D'_0\rangle + x^{-1}\langle D'_\infty\rangle = \langle D'\rangle$$

□

Proposition 1.3.4. *On the contrary, for the first Reidemeister moves we do not have the invariance*





Proof.

$$\langle D\rangle = x\langle D_0\rangle + x^{-1}\langle D_\infty\rangle = x\langle D'\rangle + x^{-1}\langle \widehat{D}_0\rangle =$$

$$x\langle D'\rangle + x^{-1}\langle \widehat{D}'\rangle = x\langle D'\rangle + x^{-1}(-x^2 - x^{-2})\langle D'\rangle =$$

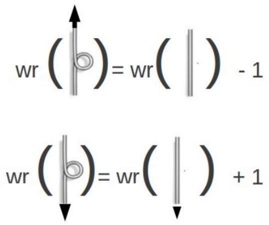
$$= (x - x - x^{-3})\langle D' \rangle = -x^{-3}\langle D' \rangle$$

□

In order to gain the invariance of the first Reidemeister move, we need a correction factor, which can be given in terms of the writhe number.

The writhe number $w(D)$ of a diagram D of an oriented link is the sum of the signs of the crossings of D, where each crossings has sign $+1$ or -1 as defined by convention in Figure 1.9.

We note that $w(D)$ does not change if D changes under the second and the third Reidemeister moves, while it does change by ± 1 if D is changed by the first Reidemeister move, as it is shown in the figure below.



Now, we can define the Kauffman polynomial:

Definition 1.3.5. *Let D be a diagram of an oriented link L. Then the Kauffman polynomial is the invariant of the oriented link L*

$$P_D(x) = (-x)^{-3w(D)}\langle |D| \rangle$$

where $|D|$ is the diagram D without the orientation.

Proposition 1.3.6. *The Kauffman polynomial is invariant under the Reidemeister moves*

Proof. We do not need to control the invariance under the second and third Reidemeister moves, because the Kauffman polynomial is composed by factors that are already invariant under such moves.

We only have to check the invariance of $P_D(x)$ under the first move, that is:

$$P_{D'}(x) = ((-x))^{-3wr(D')}\langle |D'| \rangle =$$

$$((-x))^{-3(wr(D)+1)}\frac{\langle |D| \rangle}{-x^{-3}} = ((-x))^{-3wr(D)}\langle |D| \rangle = P_D(x)$$

□

Proposition 1.3.7. *The Kauffman polynomial does not depend on the diagram of the link L.*

Definition 1.3.8. *According to the proposition above, the Kauffman polynomial of a link L is equal to the Kauffman polynomial of any possible diagram D of L. Thus, $P_D(x) = P_L(x)$.*

We define the Jones polynomial:

Definition 1.3.9.

$$V_L(t) = P_L(x^{-\frac{1}{4}}) \in \mathbb{Z}[t^{-\frac{1}{2}}, t^{\frac{1}{2}}] \ .$$

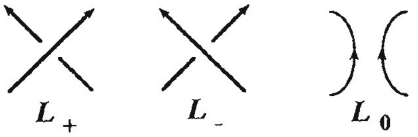
Proposition 1.3.10. *The Jones polynomial is a function*

$$V : \{\textit{Oriented links in } S^3\} \to \mathbb{Z}[t^{-\frac{1}{2}}, t^{\frac{1}{2}}]$$

uniquely determined by the following conditions:

(i) $V_{\bigcirc}(t) = 1$*;*

(ii) *whenever three oriented links L_+, L_- and L_0 are the same, except in the neighborhood of a crossing where they are as shown below*



then

$$tV_{L_+}(t) - t^{-1}V_{L_-}(t) = (t^{-\frac{1}{2}} - t^{\frac{1}{2}})V_{L_0}(t) \ . \tag{1.1}$$

Proof. Remembering the resolutions in figure (1.10), we have:

$$\langle D_+\rangle = x\langle D_0\rangle + x^{-1}\langle D_\infty\rangle \ ,$$
$$\langle D_-\rangle = x^{-1}\langle D_0\rangle + x\langle D_\infty\rangle \ .$$

We multiply the first equation by x and the second by x^{-1} and then we consider the difference

$$x\langle D_+\rangle - x^{-1}\langle D_-\rangle = (x^2 - x^{-2})\langle D_0\rangle \ .$$

Thus, using the fact that in these diagrams

$$w(D_+) - 1 = w(D_0) = w(D_-) + 1 \ ,$$

it follows that:

$$-x^4 P_{L_+}(x) + x^{-4} P_{L_-}(x) = (x^2 - x^{-2}) P_{L_0}(x) \ .$$

The substitution that we mentioned before ($x^{-\frac{1}{4}} = t$) guarantees that (1.1) holds.

The uniqueness follows by induction on the number of crossing and on the number of crossing that we need in order to get the unknot.

Base case $V_{\bigcirc}(t) = 1$;

$V_{\bigcirc \sqcup \cdots \sqcup \bigcirc}(t) = (t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{n-1}$ with $n =$ number of $\sqcup$. In that case, use the second property of 1.3.2.

Inductive step From the skein relation

$$t V_{L_+}(t) - t^{-1} V_{L_-}(t) = (t^{-1/2} - t^{1/2}) V_{L_0}(t) \ .$$

We assume that both V_{L_-} and V_{L_0} are univocally determined by the two conditions mentioned above. Then also the V_{L_+} is univocally determined by construction.

□

Property 1.3.11. *We observe that $V_L(t)$ has only even powers of t variable. More precisely if the number of the components of L is odd then all the exponents of t are $\equiv_4 0$, otherwise they are $\equiv_4 2$.*

Proof. The proof follows from the skein relation

$$t^4 V_{L_+}(t) - t^{-4} V_{L_-}(t) = (t^{-2} - t^2) V_{L_0}(t)$$

and from the normalization $V_{\bigcirc}(t) = 1$.

We observe that the number of the components is:

$$n_{c(L_0)} \pm 1 = n_{c(L_+)} = n_{c(L_-)} \ . \qquad (1.2)$$

We assume that for V_{L_-} and for V_{L_0} the property holds, i.e. the exponents of x are $\equiv_4 0$ or $\equiv_4 2$.

We want to prove that the property holds also for V_{L_+}:

$$V_{L_+}(t) = t^{-4}(t^{-4} V_{L_-}(t) + (t^{-2} - t^2) V_{L_0}(t)) \ .$$

If $n_{c(L_0)}$ is odd the exponent of the t will be, by hypothesis, $\equiv_4 0$, while, using (1.2), $n_{c(L_-)}$ is even and the exponent of the t will be $\equiv_4 2$.

Then, after the product with the coefficients associated to both V_{L_0} and V_{L_-} we can say that also the exponent of t is $\equiv_4 2$. Thus also the exponent associated to V_{L_+} will be $\equiv_4 2$.

If $n_{c(L_0)}$ is even the exponent of the t will be, by hypothesis, $\equiv_4 2$, while, using (1.2), $n_{c(L_-)}$ is odd and the exponent of the t will be $\equiv_4 0$.

Then, after the product with the coefficients associated to both V_{L_0} and V_{L_-} we can say that also the exponent of t is $\equiv_4 0$. Thus also the exponent associated to V_{L_+} will be $\equiv_4 0$.

□

In the rest of the thesis we will denote the Jones polynomial in the x variable, $V_L(x) \in \mathbb{Z}[x^{\frac{1}{2}}, x^{-\frac{1}{2}}]$, taking into account the substitution mentioned before.

Chapter 2

Braids

In this second chapter we focus on braids. After having given the basic definitions, we introduce the braid groups B_n, that will play a crucial role in the third chapter. To establish the connection between braids and knots, we define two different closures of a braid: the trace closure and the plat closure. Then, we give a different definition of the Jones Polynomial in terms of braids.

2.1 Braids

In this section we give the geometrical definition of braids. From now, we consider the Euclidean 3-space $\mathbb{R}^3$ and the portion of it between the two parallel planes with z-coordinates 0 and 1.

Definition 2.1.1. *A geometric braid b is the disjoint union of a_i, $b = \sqcup_{i=1}^n a_i$, with a_i smooth topological arcs, called strands, such that:*

1. *a_i goes from $(i, 0, 0)$ to $(\delta(b)(i), 0, 1)$ in $\mathbb{R}^2 \times [0, 1]$, where $\delta(b) \in \Sigma_n$, it goes from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, n\}$ and we call it permutation associated with the braid b ;*

2. *$a_{i_1} \cap a_{i_2} = \varnothing \ \forall i_1 \neq i_2$;*

3. *the arcs are monotonic with respect to the z-coordinate, i.e. the projection: $\pi_i : a_i \to z$-axis is differential and regular (it is bijective in $[0, 1]$).*

For each $i = 1, \dots, n$, the arc a_i admits a unique smooth parametrization $\alpha_i : [0,1] \to \mathbb{R}^2 \times [0,1]$ such that $\alpha_i(t) = (x_i(t), y_i(t), t)$, $\alpha_i(0) = (i, 0, 0)$ and $\alpha_i(1) = (\delta(b)(i), 0, 1)$. This allows us to interpret a braid as a loop in the space of all the subset of the plane consisting of n points, that is

$$\Gamma_n \mathbb{R}^2 = \{\{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathbb{R}^2\} .$$

Namely, such loop $\alpha : [0,1] \to \Gamma_n \mathbb{R}^2$ is defined by

$$\alpha(t) = \{(x_1(t), y_1(t)), \dots, (x_n(t), y_n(t))\} .$$

As we have done for knots, we are now interested to define a notion of isotopy equivalence between two braids.

Definition 2.1.2. *Two geometric braids b and b' on n-strings are isotopic if b can be continuously deformed into b' in the space of braids. That is, b and b' are isotopic by a level preserving isotopy $F : b \times \{0,1\} \to \mathbb{R}^2 \times \{0,1\}$ such that for all the $s \in \{0,1\}$, $F_s : b \to \mathbb{R}^2 \times [0,1]$, $F_s(x,y,z) = (x', y', z)$ and then $F_1(b) = b'$ and $F_0(b) = b$ with $F_0 = Id_{\mathbb{R}^2 \times [0,1]}$*

It can be seen that the relation of isotopy is an equivalence relation on the set of geometric braids on n-strings.

In accordance with the interpretation of a braid as a loop (mentioned before), the isotopy between two braids corresponds to an homotopy between loops in the configuration space.

As we have already done with knots, we can draw braids in a plane using diagrams. Up to braid isotopy we can always assume that the projection (and so the diagram) on the $x - z$ plane respects the condition below.

There is a finite number of crossing points at which exactly two strands meet: one of them is distinguished and it said to be undergoing (and the other is overgoing).

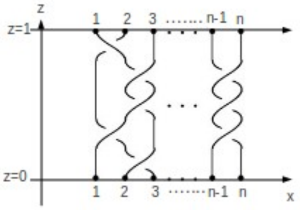


Figure 2.1: An n-braid diagram.

2.2 The Braid group, B_n

Here, we see that n-braids form a group B_n, with respect to a certain natural composition. Then, we give an algebraic description of B_n in terms of generators and relations.

Firstly, we define the product between two n-braids as follows. We put the second braid on the top of the first one and then we rescale the z-coordinate by a factor $1/2$, in order to make the union into a new braid, with the $z-$coordinate in $[0, 1]$. As an example, let see:

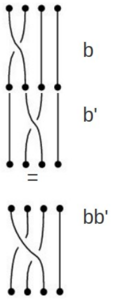


Figure 2.2: The composition of b and b'.

This definition of the product operation for braids is compatible with the isotopy relation, hence the *braid product* induces a (well defined) product between isotopy classes of braids.

Furthermore, we can prove that this is a group operation. In fact, the set B_n and the product $\cdot$ satisfy the four requirements know as the group axioms.

The identity element consists of an n-braid with 0-crossings, while the inverse element of a braid $a \in B_n$ is $b = \tau(a)$ where $\tau : \mathbb{R}^2 \times [0,1] \to \mathbb{R}^2 \times [0,1]$ is the function that upside down the braid, inverting the arcs from the top to the bottom, so

$$\tau(x,t) = (x, 1-t) \ .$$

We also observe that all these properties are valid up to isotopy.

We are now able to define the Braid group.

Definition 2.2.1. *B_n is the group the the isotopy classes of the $n-$braids with the product defined above.*

Braid groups were introduced explicitly by Emil Artin in 1925 (from which the name *Artin Braid group*) although they were already implicit in a work by Adolf Hurwitz (1891). According to the interpretation of the braids as loops in the space of all the subsets of the plane $(\Gamma_n \mathbb{R}^2)$, Hurwitz defines the Braid group as the fundamental group of the configuration space.

Theorem 2.2.2. *The braid group B_n admits a presentation with $n-1$ generators $b_1, b_2, \ldots, b_{n-1}$ and the braid relations*

$$b_i b_j = b_j b_i \quad for \quad |i-j| \geq 2 \ ;$$

$$b_i b_{i+1} b_i = b_{i+1} b_i b_{i+1} \forall i = 1, 2, \ldots, n-2 \ .$$

Proof. See [9]. □

Forgetting how the strands twist and cross, every braid determines a permutation on n elements. The group of such permutations is Σ_n It has three families of relations:

1. $t_i t_{i+1} t_i = t_{i+1} t_i t_{i+1} \forall i$;
2. $t_i t_j = t_j t_i \forall i, j : |i-j| > 1$;
3. $t_i^2 = 1 \forall i$.

We observe that the first two families of relations are exactly the same of the families of relations that we have in B_n. Thus

$$\delta : B_n \to \Sigma_n$$

define a surjective group homomorphism from the braid group to the symmetric group.

2.3 Closed braid

Now, we are interested to associate to any given n-braid a link, by connecting the endpoints of the arcs $a_1, \dots, A_n$.

We consider two different ways to do that: the trace closure and the plat closure. Actually, while the former is defined for any n, the latter is defined only when n is even.

Definition 2.3.1. *The **trace** closure of a braid b corresponds to the link obtained by connecting, one by one, all the strands at the top to the corresponding strands at the bottom, with parallel arcs on the right of the braid that do not meet each other. We denote it b^{tr}:*

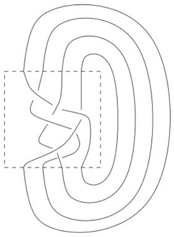


Figure 2.3: An example of the trace closure of a 4-strand braid.

Definition 2.3.2. *The plat closure of a $2n-$strand braid b corresponds to a link obtained by connecting in pairs adjacent endpoints of the arcs $a_1, \dots, A_n$ on the bottom and on the top of the braid. We call it b^{pl}:*

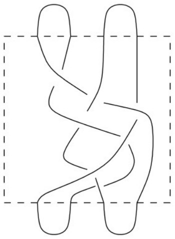


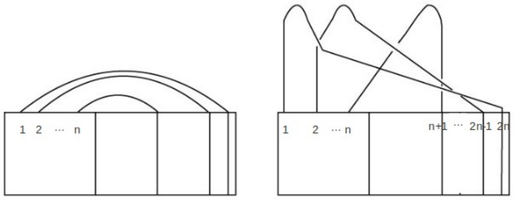
Figure 2.4: An example of the plat closure of a 4-strand braid.

As we know from the Alexander's theorem (1923):

Theorem 2.3.3. *For every link L there exists some braid $b \in B_n$ such that the trace closure of b is L.*

Proof. See [11]. Actually, we will use the algorithmic proof given by P.Vogel. Basically he shows that every link diagram can be transformed into a closed braid by a sequence of II Reidemeister moves. □

The trace closure of a n-braid is isotopic to the plat closure of a $2n$-braid, as it can be seen in figure:



And we have the following analogue of Alexanders theorem

Theorem 2.3.4. *For every link L there exists some braid $b \in B_{2n}$ such that the plat closure of b is L.*

Finally, in order to know when the trace closure of two braids represent the same knot (or link), we use the Markov moves.

Definition 2.3.5. *Let β be a $n-$braid, $\beta \in B_{n+1}$,*

first type *let γ be another generic $n-$braid,*

$$\beta \to \beta' = \gamma\beta\gamma^{-1} \ ;$$

second type *let b_n be the generator of the B_{n+1} group,*

$$\beta \to \beta' = \beta b_n \ , \quad \beta' = \beta b_n^{-1} \ .$$

Theorem 2.3.6. *Let b and b' be two (oriented) braids not necessarily with the same number of strings. The trace closure of b and b' represents the same (oriented) knot (or link) K if and only if b can be deformed in b' with a finite number of Markov moves (or their inverses), i.e. it exists a finite sequence*

$$b = b_0 \to \cdots \to b_m = b'$$

such that, for $i = 0, \dots, m-1$, b_{i+1} is obtained from b_i by a Markov move.

2.4 Hecke Algebra

In this section we introduce the algebraic structure needed to give an algebraic description of the Kauffman polynomial and the Jones polynomial based on braids.

Definition 2.4.1. *Let $\mathbb{K}$ be a field and $q \in \mathbb{K} - \{0\}$. $\forall n \geq 1$, $\mathbb{H}_n$ is the $\mathbb{K}$-algebra generated by $t_1, t_2, \ldots, t_{n-1}$ with the relations:*

$$t_i t_{i+1} t_i = t_{i+1} t_i t_{i+1} \quad \forall i$$

$$t_i t_j = t_j t_i \quad \forall i, j : |j - i| > 1$$

$$t_i^2 = (q-1)t_i + q \quad \forall i$$

$\mathbb{H}_n$ is an associative (but not commutative) algebra. It can be seen as the free algebra generated by the two-sided ideal generated by the relations above.

We call $\mathbb{H}_n$ Hecke Algebra.

We observe that for every $n \geq 1$ there is a natural inclusion

$$i_n : \mathbb{H}_{n-1} \subset \mathbb{H}_n \ .$$

Proposition 2.4.2. *There exists an isomorphism*

$$\phi : \mathbb{H}_n \oplus \mathbb{H}_n \otimes_{\mathbb{H}_{n-1}} \mathbb{H}_n \to \mathbb{H}_{n+1} \tag{2.1}$$

$$\phi(a, \sum_i b_i \otimes c_i) = a_n + \sum_i b_i t_n c_i$$

($a_n, b_i, c_i \in \mathbb{H}_n$ while $t_n \in \mathbb{H}_{n+1}$).

Proof. The proof is divided into four steps.

1. ϕ is well defined.

   Let $u \in \mathbb{H}_{n-1}$, thus it is a linear combination of monomials in $t_1, \ldots, t_{n-2}$ that commute with t_n in $\mathbb{H}_{n+1}$. We want to check if $\phi(bu \otimes c) = \phi(b \otimes uc)$.

   $$\phi(bu \otimes c) = but_n c \ , \quad \phi(b \otimes uc) = bt_n uc$$

   Hence

   $$but_n c = bt_n uc \ ,$$

   because all the components of u commute with t_n. Thus, ϕ is well defined.

2. ϕ is surjective.

   We have to show that $\mathbb{H}_{n+1}$ is generated as a vector space on $\mathbb{K}$ by the monomials with, at most, one t_n.

   By induction on n and on the number of t_n's, let have two occurrence of t_n, thus $M = M_1 t_n M_2 t_n M_3$ where M_2 is a monomial in $t_1, \ldots, t_{n-1}$.

   - If M_2 does not contain t_{n-1}:

     $$M = M_1 M_2 t_n^2 M_3 = (q-1) M_1 M_2 t_n M_3 + q M_1 M_2 M_3 \ ,$$

     and thus we decrease the number of t_n's.
   - If M_2 contains one t_{n-1}, thus $M_2 = M' t_{n-1} M''$, where M', M'' are two monomials in $t_1, \ldots, t_{n-2}$ that commute with t_n.

     $$M = M_1 t_n M' t_{n-1} M'' t_n M_3 = M_1 M' t_n t_{n-1} t_n M'' M_3 = M_1 M' t_{n-1} t_n t_{n-1} M'' M_3 \ ,$$

     and thus we decrease the number of t_n's.

   Then, any element of $\mathbb{H}_{n+1}$ is a sum $a + \sum_i b_i t_n c_i$, with $a, b_i, c_i \in \mathbb{H}_n$. Thus ϕ is surjective.

3. Monomial in normal form generate $\mathbb{H}_{n+1}$ over $\mathbb{K}$.

   Let consider the lists of monomials:

   - $S_1 = \{1, t_1\}$,
   - $S_2 = \{1, t_2, t_2 t_1\}$,
   - $S_3 = \{1, t_3, t_3 t_2 t_1\}$,
   - $\ldots$
   - $S_n = \{1, t_n, t_n t_{n-1}, \ldots, t_n t_{n-1} \cdots, t_1\}$

   and

   $$v_i \in S_i \Rightarrow t_{i+1} v_i \in S_{i+1} \ .$$

   We call monomial in normal form the monomial $M = u_1 \cdots u_n$ for all possible choices of $u_i \in S_i$, for $i = 1, \ldots, n$; they are $(n+1)!$.

   We prove that this monomials M generate $\mathbb{H}_{n+1}$ as a $\mathbb{K}$-space. Consequently:

$$dim_K \mathbb{H}_{n+1} \leq (n+1)! \tag{2.2}$$

$$dim_K \{\mathbb{H}_n \oplus \mathbb{H}_n \otimes_{\mathbb{H}_{n-1}} \mathbb{H}_n\} \leq (n+1)! \tag{2.3}$$

We may assume that the monomials M generate $\mathbb{H}_n$ as a $\mathbb{K}$-space. We want to check if this holds also for $\mathbb{H}_{n+1}$.

$\mathbb{H}_{n+1}$ is generated on $\mathbb{K}$ by the monomials M_0 and $M = M_1 t_n M_2$ with M_0, M_1, M_2 monomials in $t_1, \ldots, t_{n-1}$.

- For M_0 there is a trivial proof;
- For $M = M_1 t_n M_2$: let M_2 be a $\mathbb{K}$-linear composition of monomials $v_1, v_2, \ldots, v_{n-1}$ with $v_i \in S_i$ for $i = 1, \cdots, n-1$.

  $$M_1 t_n v_1 \cdots v_{n-1} = M_1' t_n v_{n-1} = m_1' u_n \ .$$

  Also M_1' is a linear composition of monomials of the forms $u_1, \ldots, u_{n-1}$ with $u_i \in S_i$ for $i = 1, \cdots, n-1$.

Thus, M is a linear combination of monomials $u_1 \cdot u_2 \cdots u_n$ as we want and (2.2) holds. This also shows that $\mathbb{H}_n \otimes_{\mathbb{H}_{n-1}} \mathbb{H}_n$ is spanned over $\mathbb{K}$ by the subspace $\mathbb{H}_n \otimes u_{n-1}$ with $u_{n-1} \in S_{n-1}$. Therefore (2.3) holds.

4. Monomials having the normal form $M = u_1 \cdots u_n$ with $u_i \in S_i$ for $i = 1, \ldots, n-1$ are $\mathbb{K}$- linearly independent and hence ϕ is an isomorphism.

   Let Σ_{n+1} be symmetric group on $\{1, \ldots, n+1\}$, and s_i be the transposition $(i, i+1)$. Thus $\pi \in \Sigma_{n+1}$ is of the form $\pi = w_1 \cdots w_n$ with $w_i \in \{1, \ldots, s_i s_{i-1} \cdots s_1\}$.

   We define $l : \Sigma_{n+1} \to N$ as the word length function in Σ_{n+1} relative to the generators $\{s_1, s_2, \ldots, s_n\}$. For $i = \{1, \ldots, n\}$ we define $L_i \in End_{\mathbb{K}}(\mathbb{K}\Sigma_{n+1})$ with

   $$L_i(\pi) = \begin{cases} s_i\pi, & \text{if } l(s_i\pi) > l(\pi) \\ q s_i \pi + (q-1)\pi, & \text{if } l(s_i\pi) < l(\pi) \end{cases}$$

   The crucial fact is that there exists an algebra map

   $$L : \mathbb{H}_{n+1} \to End_{\mathbb{K}}(\mathbb{K}\Sigma_{n+1})$$

such that $L(t_i) = L_i$ for $i = 1, \dots, n$. We assume that the endomorphisms $L_i \in End_{\mathbb{K}}(\mathbb{K}\Sigma_{n+1})$ satisfy the defining relations in definition (2.4.1).

Let now consider a monomial in normal form $M = u_1 \cdots u_n$ such that $u_i = t_i t_{i-1} \cdots t_{i-j}$, then

$$L(M) : 1 \mapsto w_1 \cdots w_n$$

where $w_i = s_i s_{i-1} \cdots s_{i-j}$.

We already know that all the $(n+1)!$ elements of Σ_{n+1} are of the form $w_1 \cdots w_n$. Such elements are $\mathbb{K}$-linearly independent in $\mathbb{K}\Sigma_{n+1}$.

Thus, as the map from $\mathbb{H}_{n+1}$ to $\mathbb{K}\Sigma_{n+1}$ ($x \to L(x)(1)$) is $\mathbb{K}$- linear, the elements $M = u_1 \cdots u_n$ in the normal form are linearly independent and $dim_{\mathbb{K}}(\mathbb{H}_{n+1}) = (n+1)!$.

A dimension count prove that the surjective map ϕ is an isomorphism.

□

The fundamental idea of Jones, which led him to the definition of his polynomial, is the construction of the trace function Tr_n described below.

Definition 2.4.3. *A trace is a linear function from an algebra A to $\mathbb{C}$*

$$tr : A \Rightarrow \mathbb{C}$$

that satisfies $tr(XY) = tr(YX)$ for every two elements X, Y in the algebra.

Theorem 2.4.4. *For every $z \in \mathbb{K}$ there exists a unique family of applications $\mathrm{Tr}_n : \mathbb{H}_n \to \mathbb{K}$ which are $\mathbb{K}$-linear but not algebra homomorphisms, i.e. they do not respects the algebra product, and they satisfy the below properties:*

1. *this diagram commutes:*

$$\begin{tikzcd} H_n \arrow[rr, "i_{n+1}"] \arrow[dr, "\mathrm{Tr}_n"'] & & H_{n+1} \arrow[dl, "\mathrm{Tr}_{n+1}"] \\ & K & \end{tikzcd}$$

2. $\mathrm{Tr}_n(1) = 1$*;*

3. $\mathrm{Tr}_n(xy) = \mathrm{Tr}_n(yx)$*;*

4. $\mathrm{Tr}_{n+1}(xt_ny) = z\,\mathrm{Tr}_n(xy)\ \forall x, y \in \mathbb{H}_n$

Proof. See [5]. □

2.5 Jones polynomial.

In section 2.2 we saw that there exists a homomorphism between B_n, the braid group, and Σ_n, the permutations group.

Our purpose is to find a function from B_n to an algebra $\mathbb{K}$. Remembering that in the section 2.4 we have a family of applications

$$\mathrm{Tr}_n : \mathbb{H}_n \to \mathbb{K}\ ,$$

it will suffice to find a function ρ_n from B_n to $\mathbb{H}_n$, and then take the composition: for $q \in \mathbb{C}$ and $z \in \mathbb{Z}$

$$V_{q,z} : B_n \xrightarrow{\rho_n} \mathbb{H}_n \xrightarrow{\mathrm{Tr}_n^z} \mathbb{K}(q, z)\ .$$

In that way $b \in B_n$ will become $V_{q,z}(b) = v_b(q, z) \in \mathbb{K}[q, q^{-1}, z, z^{-1}] \subset \mathbb{K}(q, z)$.

Given $q \in \mathbb{C}$ and $z \in \mathbb{Z}$, for every braid b we can define:

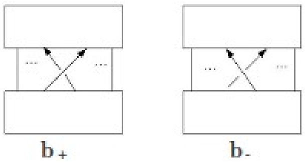
$$W_b(q, z) = \left(\frac{1}{z}\right)^{(n(b)+\gamma(b)-1)/2} \left(\frac{q}{z-q+1}\right)^{(n(b)-\gamma(b)-1)/2} V_b(q, z)$$

where $n(b)$ is the number of the strands of b and $\gamma(b)$ is the number of the crossings.

$W_b(q, z) \in \mathbb{K}[q^{\pm\frac{1}{2}}, z^{\pm\frac{1}{2}}]$ and $W_{S^1}(q, z) = 1$.
Recalling the last construction of the map V_b,

$$W_b(q, z) = \mathrm{Tr}\left(\left(\frac{1}{z}\right)^{(n(b)+\gamma(b)-1)/2} \left(\frac{q}{z-q+1}\right)^{(n(b)-\gamma(b)-1)/2} \rho(b)\right)$$

Remembering that



the invariant will be:

$$W_{b_\pm}(q,z) = \mathrm{Tr}\left(\left(\frac{1}{z}\right)^{(n(b_\pm)+\gamma(b_\pm)-1)/2}\left(\frac{q}{z-q+1}\right)^{(n(b_\pm)-\gamma(b_\pm)-1)/2}\rho(b_\pm)\right)$$

In order to simplify the writing, we say that

$$W_b(q,z) = \mathrm{Tr}(c\rho(b)) = c\,\mathrm{Tr}(\rho(b)) \ ;$$

$$W_{b_+} = \mathrm{Tr}(c\left(\tfrac{1}{z}\right)^{\frac{1}{2}}\left(\tfrac{q}{z-q+1}\right)^{-\frac{1}{2}}\rho(b_+)) \ ;$$

$$W_{b_-} = \mathrm{Tr}(c\left(\tfrac{1}{z}\right)^{-\frac{1}{2}}\left(\tfrac{q}{z-q+1}\right)^{\frac{1}{2}}\rho(b_-)) \ .$$

Now, we multiply W_{b_+} and W_{b_-} for two appropriate coefficients:

$$\left(\tfrac{z}{z-q+1}\right)^{\frac{1}{2}} W_{b_+} = c\,\mathrm{Tr}(q^{-\frac{1}{2}}\rho(b_+))$$

$$\left(\tfrac{z}{z-q+1}\right)^{-\frac{1}{2}} W_{b_+} = c\,\mathrm{Tr}(q^{\frac{1}{2}}\rho(b_-)).$$

Subtract the former to the latter:

$$\left(\frac{z}{z-q+1}\right)^{\frac{1}{2}} W_{b_+} - \left(\frac{z}{z-q+1}\right)^{-\frac{1}{2}} W_{b_+} = c\,\mathrm{Tr}(q^{-\frac{1}{2}}\rho(b_+)) - q^{\frac{1}{2}}\rho(b_-)). \tag{2.4}$$

Taking into account the expression of the $\rho(b_{\pm})$:

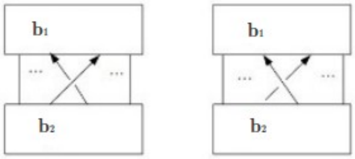


Figure 2.5: $\rho(b_+) = \rho(b_1)t_i\rho(b_2) \qquad \rho(b_-) = \rho(b_1)t_i^{-1}\rho(b_2)$.

we can write that

$$c\,\mathrm{Tr}(q^{-\frac{1}{2}}\rho(b_+)) - q^{\frac{1}{2}}\rho(b_-)) = c\,\mathrm{Tr}\left(\rho(b_1)(q^{-\frac{1}{2}}t_i - q^{\frac{1}{2}}t_i^{-1})\rho(b_2)\right) \ .$$

First, let us compute $(q^{-\frac{1}{2}}t_i - q^{\frac{1}{2}}t_i^{-1})$. We have

$$(q^{-\frac{1}{2}}t_i - q^{\frac{1}{2}}t_i^{-1}) = \left(q^{-\frac{1}{2}}t_i - q^{\frac{1}{2}}\left(\frac{t_i}{q} - \frac{q-1}{q}\right)\right) = \frac{q^{\frac{1}{2}}(q-1)}{q} = q^{\frac{1}{2}}(1-q^{-1}) = q^{\frac{1}{2}} - q^{-\frac{1}{2}}$$

Thus, we get

$$c\,\mathrm{Tr}((q^{\frac{1}{2}} - q^{-\frac{1}{2}})\rho(b_1)\rho(b_2)) = c\,\mathrm{Tr}((q^{\frac{1}{2}} - q^{-\frac{1}{2}})\rho(b))$$

$$c(q^{\frac{1}{2}} - q^{-\frac{1}{2}})\,\mathrm{Tr}(\rho(b)) = (q^{\frac{1}{2}} - q^{-\frac{1}{2}})W_b(q,z)$$

And so, the expression [2.4] becomes

$$\left(\frac{z}{z-q+1}\right)^{\frac{1}{2}} W_{b_+} - \left(\frac{z}{z-q+1}\right)^{-\frac{1}{2}} W_{b_+} = (q^{\frac{1}{2}} - q^{-\frac{1}{2}})W_b(q,z) \ .$$

Now we can define the two polynomials. First of all we change (for convenience) the variables:

$$x = \left(\frac{z}{z-q+1}\right)^{\frac{1}{2}} \qquad y = q^{\frac{1}{2}} - q^{-\frac{1}{2}} \ .$$

The **Kauffman Polynomial** is:

$$P_K(x,y) = W_\beta(q,z) \quad \text{and} \quad x^{-1}P_{K+}(x,y) - xP_{K-}(x,y) = yP_K(x,y).$$

For $y = x^{\frac{1}{2}} - x^{-\frac{1}{2}}$ we obtain the **Jones Polynomial** in two variables and its characteristic equation is

$$x^{-1}V_{K+}(x) - xV_{K-}(x) = (x^{\frac{1}{2}} - x^{-\frac{1}{2}})V_K(x) \ .$$

Chapter 3

Temperley Lieb Algebras

In this third chapter we study the Temperley-Lieb algebras $TL_n(d)$. These are the algebraic ambient where we will develop the quantum algorithm for the approximation of the Jones Polynomial. Indeed, we will see that there exists a representation of B_n in the $TL_n(d)$ algebra and that the Jones Polynomial can be defined as a certain trace function on the image of the braid group in $TL_n(d)$. Moreover this trace has an additional property (Markov property) that makes it unique. So, in the end of the chapter, our goal will be the approximation of this trace function (see Chapter 5).

3.1 The TL_n Algebra

The starting point of this chapter is the Temperley-Lieb algebra, which has played a central role in the discovery by Vaughan Jones of his new polynomial invariant of knots and links and in the subsequent developments of knot theory over the past two decades.

We begin with the algebraic presentation:

Definition 3.1.1. *Let $n \in \mathbb{Z}$ and $d \in \mathbb{C}$. The Temperley-Lieb Algebra $TL_n(d)$ is the algebra generated by $1, e_1, \dots, e_{n-1}$ with the relation:*

$e_i e_j = e_j e_i$, $|i-j| \geq 2$;

$e_i e_{i\pm1} e_i = e_i$;

$e_i^2 = de_i$.

In order to describe geometrically this algebra we will use the Kauffman n-diagrams:

Definition 3.1.2. *Let R_n be a rectangle with n marked boundary points on the top edge and on the bottom edge. A Kauffman n-diagram is a picture draw inside R_n consisting of n non-intersecting curves that begin and end at distinct marked boundary points.*

We consider two diagrams equal if they are isotopically equivalent keeping the boundary fixed.

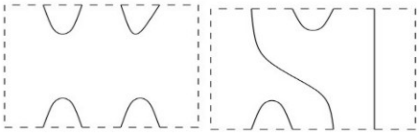


Figure 3.1: Examples of Kauffman n-diagrams.

Definition 3.1.3. *$gTL_n(d)$ is the vector space formed by the linear combination of Kauffman diagrams and coefficients in $\mathbb{C}$.*

Then, these Kauffman diagrams are the basis of the vector space $gTL_n(d)$.

In order to define the Algebra structure, we describe the operation of product that we use in the $TLn(d)$ algebra.

This operation can be separate into two parts:

1. we concatenate the two diagrams as we did for braids (one on the top of the other);

2. we replace the k closed components with a proper coefficient $d^k \in \mathbb{C}$.

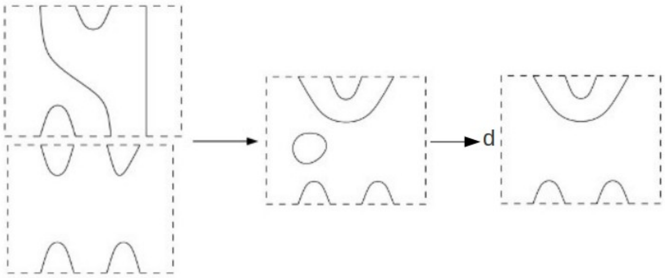
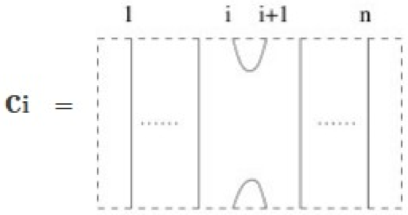


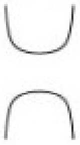
Figure 3.2: An example of the multiplication rule.

Thus, after extending the multiplication rule to all the elements we can obtain another algebra $gTL_n(d)$.

Theorem 3.1.4. *The map $\psi : TL_n(d) \to gTL_n(d)$, defined by $\psi(e_i) = c_i$, is an isomorphism of algebras.*



Such c_i are calles *capcups*:



Property 3.1.5. *Each Kauffman diagram can be written as the product of c_i diagrams, thus these c_i, for $i = 1, \dots, n-1$, are the generators of the algebra $gTL_n(d)$.*

Proof. See [10]. □

3.2 Representing B_n into TL_n

We define a map from the braid group to the Temperley-Lieb Algebra.

Definition 3.2.1. *For each $a \in \mathbb{C}$ such that $d = -a^{-2} - a^2$ we define $\rho_d : B_n \to TL_n(d)$ such that*

$$\rho_d(b_i) = ae_i + a^{-1}1 \ . \tag{3.1}$$

$\forall b_i$ generator of B_n and $\forall e_i$ generator of $TL_n(d)$.

Property 3.2.2. *The mapping that we have just create is a representation of B_n in $TL_n(d)$.*

Proof. We have to check if the relation of the braid group are satisfied by ρ_d.

- for $|i - j| > 1$, $\rho_d(b_i)$ commutes with $\rho_d(b_j)$ since e_i commutes with e_j (see relation 1 of the TL_n);
- $$\rho_d(b_i)\rho_d(b_{i+1})\rho_d(b_i) = \rho_d(b_{i+1})\rho_d(b_i)\rho_d(b_{i+1}) :$$

  $\rho_d(b_i)\rho_d(b_{i+1})\rho_d(b_i) = a^3e_ie_{i+1}e_i + ae_{i+1}ei + ae_i^2 + a^{-1}e_i + ae_ie_{i+1} + a^{-1}e_{i+1} + a^{-1}e_i + a^{-3}$

  $\rho_d(b_{i+1})\rho_d(b_i)\rho_d(b_{i+1}) = a^3e_{i+1}e_ie_{i+1} + ae_ie_{i+1} + ae_{i+1}^2 + a^{-1}e_{i+1} + ae_{i+1}e_i + a^{-1}e_i + a^{-1}e_{i+1} + a^{-3}$.

  Since $d = -a^{-2} - a^2$,

  $$a^{-1} + ad + a^3 = a^3 + (-a^{-2} - a^2)a + a^{-1} = a^3 - a^3 - a^{-1} + a^{-1} = 0$$

  Then, after removing equal terms and applying the relations of the $TL_n(d)$, the wanted equality becomes

  $$(a^{-1} + ad + a^3)e_i = (a^{-1} + ad + a^3)e_{i+1}. \tag{3.2}$$

□

Let τ be a linear representation of $TL_n(d)$, we will use the representation given by the ρ_d to derive a linear representation of B_n by composition. Thus, we define the map ϕ by specifying its action on the generators b_i of B_n:

$$\phi(b_i) = \phi_i = \tau(\rho_d(b_i)) = a\tau(e_i) + a^{-1}1 \ .$$

Property 3.2.3. *If $|a| = 1$ and $\tau(e_i)$ are Hermitian for all i, the map ϕ is a unitary representation of B_n.*

Proof.

$$\tau(\rho_d(b_i))\tau(\rho_d(b_i))^\dagger = (a_{-1}I + a\tau(e_i))((a_{-1})^*I + a^*\tau(e_i)^\dagger) =$$

$$I + a^{-2}\tau(e_i) + a^2\tau(e_i) + d\tau(e_i) = I \ .$$

□

3.3 Markov trace

Definition 3.3.1. *The Markov trace is a linear function from the algebra $gTL_n(d)$ to $\mathbb{C}$*

$$tr : gTL_n(d) \to \mathbb{C} \ .$$

It is a trace function (see definition 2.4.3) uniquely determined by this property:

if $X \in TL_{n-1}(d)$ then $tr(Xe_{n-1}) = \frac{1}{d}tr(X)$: the Markov property.

Definition 3.3.2. *The function $tr : TL_n(d) \to \mathbb{C}$, defined as*

$$tr(K) = d^{a-n}$$

is the Markov trace.

Where n is the the top and the bottom labelled points of K connected with non-intersecting curves and a is the number of loops of the resulting diagram. We can extend tr to all of $gTL_n(d)$ by linearity.

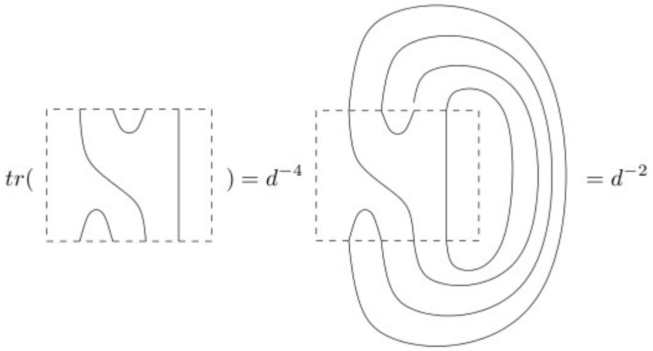
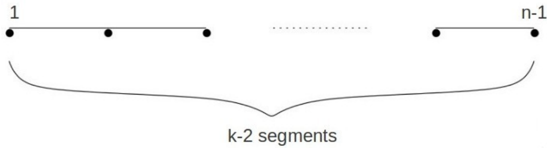


Figure 3.3: An example of Markov trace on a Kauffman diagram.

3.4 Path model representation

Now, we want to describe also a different representation of $TL_n(d)$ defined for $d = 2\cos(\pi/k)$ (due to Jones himself): the path model representation. Let $k \in \mathbb{Z}$ and G_k be the straight line graph with $k-1$ vertices and $k-2$ segments.



Let $Q_{n,k}$ be the set of all the paths of length n on the graph G_k starting from the leftmost vertex.

Given $q \in Q_{n.k}$, we describe it by a sequence of vertices of G_k :
$q(0), q(1), \ldots, q(n-1)$, where $q(i)$ in the adjacent vertex of $q(i+1)$.

We take the vector space $\mathcal{V}_{n,k}$ consisting of the linear combinations of the elements of $Q_{n,k}$, and these become a basis elements of the vector space.

Now we construct the path model representation

$$\tau : TL_n(d) \longrightarrow End(V_{n,k})$$

In that way, $x \in TL_n(d)$ becomes $\tau(x) : \mathcal{V}_{n,k} \to \mathcal{V}_{n,k}$.

We take a Kauffman n-diagram T and the description of $\tau(T)$ is given by the matrix entry $\tau(T)_{q',q}$ for each pair of $q, q' \in Q_{n,k}$.

To do this we look at the diagram and we consider the regions in which the rectangle is separated by the strands. We label the interval in which the bottom and the top edge are divided by.

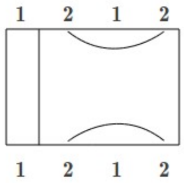
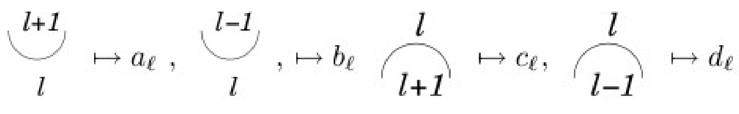


Figure 3.4: An example of labelled Kauffman diagram.

The n marked points of the Kauffman diagram divide the top and the bottom boundary into $n+1$ segments (called *gaps*).Two gaps that bound the same region in the diagram are called *connected*.

Definition 3.4.1. *We say that the pair (q', q) is* ***compatible*** *with T if, once label the gaps on the bottom from left to right by $q(0), q(1), \ldots, q(n)$ and label the gap on the top from left to right by $q'(0), q'(1), \ldots, q'(n)$, then any two connected gaps are labelled by the same vertex of G_k. Thus, we can associate the label with the region.*

The *matrix entry* $\tau(T)_{q',q}$ will only be *nonzero* if the pair of paths (q', q) is compatible with T. To each local maximum and minimum of the Kauffman diagram T, we associated a complex number as follows (it depends on the label of the regions up and down the diagram):



The matrix element $\tau(T)_{q',q}$, for a compatible pair (q', q), is defined as the product of the appropriate complex numbers over all local maxima and minima in T.

In order to have this $\tau(T)$ to be well defined, we need to show that it is invariant under isotopy of Kauffman diagrams.

In order to have an isotopic move we have to create, or eliminate, local maxima or minima in pairs.

Proposition 3.4.2. *The necessary and sufficient condition for a map to be well defined is:*

$$a_{\ell-1}d_\ell = 1 = b_{\ell+1}c_\ell. \tag{3.3}$$

Moreover, if we want to produce a representation of $TL_n(d)$, the coefficients a_ℓ, b_ℓ, c_ℓ and d_ℓ have to satisfy the equation (3.3) and

$$d = a_\ell c_\ell + b_\ell d_\ell \tag{3.4}$$

Proof. To proof (3.4) we need to verify that the $\tau(e_i)$ matrices satisfy the relations mentioned in 3.1.1, so they the matrix elements of both sides of the equalities have to be equal. Apply the operator τ to the product element is equal to apply τ to the single element that occurr in the product itself.

In the first two relations no loops are created when the operators are multiplied. The verification follows from the isotopy invariance of τ (3.3).

The third relation follows from (3.4), using that a loop is created and there are only two possible ways to label the region. □

We would like to have the $\tau(e_i)$ to be Hermitian: we add the further equation:

$$a_\ell = c_\ell^*, \quad b_\ell = d_\ell^*. \tag{3.5}$$

Then, after solving the equations (3.3-3.5), we can derive the definition of τ.

Proposition 3.4.3. *Define $\lambda_\ell = \sin\left(\frac{\pi\ell}{k}\right)$ for $\ell \in \{1, \dots, k-1\}$. Then $a_\ell = c_\ell^* = \sqrt{\frac{\lambda_\ell}{\lambda_{\ell-1}}}$ and $b_\ell = d_\ell^* = \sqrt{\frac{\lambda_\ell}{\lambda_{\ell+1}}}$ satisfy equations (3.3-3.5), with $d = 2\cos\left(\frac{\pi}{k}\right)$.*

Using these coefficients, we have the definition of $\tau(e_i)$:

Definition 3.4.4. *$\tau(e_i)_{q,q'} = 0$ if (q, q') is not compatible with e_i.*

otherwise, for a compatible pair (q, q'), $\tau(e_i)_{q,q'}$ is the product of two coefficients (the maximum and the minimum in e_i).

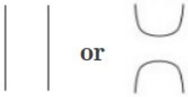
Now, having the definition of $\tau(e_i)$ we can extend it to $TL_n(d)$. Finally, for $a = ie^{-\pi/2k}$ and $\tau(e_i)$ Hermitian. We define:

Definition 3.4.5. *The unitary path model representation of B_n is defined to be $\varphi(b) = \tau(\rho_d(b))$.*

3.5 The Jones Polynomial

In the context of $gTL_n(d)$ the definition of the Kauffman and the Jones polynomials can be reviewed in another way.

Consider a link L that corresponds to the trace closure of a braid b. All the crossings are only inside b and each crossing resolution can be :



Theorem 3.5.1.

$$P_{b^{tr}}(a) = (-a)^{3w(b^{tr})} d^{n-1} tr(\rho_d(b)) \ .$$

Proof. The Jones Polynomial of an oriented link L is

$$V_L(a^{-4}) = V_{b^{tr}}(a^{-4}) = P_{b^{tr}}(a) = (-a)^{3w(L)} \langle L \rangle \tag{3.6}$$

where $w(L)$ is the writhe of the oriented link L and $\langle L \rangle$ is the bracket state sum of L (ignoring the orientation).

Using this equation we need to prove that $\langle b^{tr} \rangle = tr(\rho_d(b)) d^{n-1}$.

There exists a bijective correspondence between states that appear in the bracket sum $\langle b^{tr} \rangle$ and the Kauffman $n-$diagrams that appear in $\rho_d(b)$.

The weight of an element in the bracket state sum that corresponds to the state σ is $a^{\sigma^+ - \sigma^-} d^{|\sigma|-1}$.

The corresponding Kauffman $n-$diagram appears in $\rho_d(b)$ with the weight $a^{\sigma^+ - \sigma^-}$.

It remain to show that, for each σ, the trace of the Kauffman $n-$diagram corresponding to σ, times d^{n-1}, equals to the remaining factor in the contribution of σ to the bracket state sum, $d^{|\sigma|-1}$. That is true since the definition of the trace of a Kauffman diagram is exactly $d^{|\sigma|-n}$.

□

Chapter 4

Quantum computation

"*I think I can safely say that no one understand Quantum Mechanics.*" Feymann

Quantum computing is a new approach to computation based on Quantum Mechanics. Nowadays, efficient quantum algorithms have been discovered; they are algorithms for problems that were suppose not be treatable (in a classical sense). It is obvious that the implementations of such algorithms require quantum computer, but, at the moment, these do not exist.

This area was developed in the last years of the twentieth century and has to be thought as a new approach to the computation based on the observation that "*Information is physical*" (Landauer).

As we will see, the information will be codified by physical systems and will be elaborated by physical operations. Therefore we cannot prescind from the physical laws of the Quantum Mechanic.

4.1 Algebra background

Quantum mechanic is based on linear algebra. The formal structure of quantum mechanics is due to Dirac and Neumann. By using this formulation, we know that a state of a physical system is identified with a ray in a finite dimensional Hilbert space $\mathcal{H}$. In the finite dimensional complex vector space a Hilbert space is exactly the same thing as an inner product space.

In this section we briefly recall some basic algebraic notions and results. We refer to [6] and [7] for further details.

Consider a n-dimensional Hilbert space $\mathcal{H}$ that is a complex inner product space, i.e. $\mathcal{H}$ is a complex vector space on which there is a inner product associating to each pair of elements of $\mathcal{H}$. Fixed a orthonormal basis $|1\rangle, \dots, |n\rangle$, we can write $|v\rangle$ and we can think that this $|v\rangle$ is a column vector of $\mathcal{H}$:

$$|v\rangle = \sum_i v_i |i\rangle \quad v_i \in \mathbb{C} \ .$$

$\langle v|$, instead, indicates the linear form on $\mathcal{H}$. It can be seen as the dual vector in respect to the scalar product

$$\langle v|(|v'\rangle) = \langle v|v'\rangle \ .$$

So,regarding the dual basis, it denotes the row vector of the space $\mathcal{H}$.

In 1930 Dirac introduced this notation, and we used to call $\langle \cdot|$ *bra* and $|\cdot\rangle$ *ket.* $\langle v|v'\rangle$ is the inner product that operates in Hilbert space $\mathcal{H}$. In fact, the state of a physical system is identified with a ray in the complex separable Hilbert space, $\mathcal{H}$.

A convenient way to define linear operators on $\mathcal{H}$ is given by the outer product. Let V and W be to vector spaces, $|v\rangle \in V$ and $|w\rangle \in W$, we define the outer product $|w\rangle\langle v|$ as a linear operator from V to W such that, for all the $|v'\rangle \in V$ we have

$$(|w\rangle\langle v|)(|v'\rangle) = |w\rangle\langle v|v'\rangle = \langle v|v'\rangle|w\rangle$$

where $\langle v|v'\rangle$ is a complex number. Let $|i\rangle$ be an orthonormal basis of V such that $|v\rangle = \sum_i v_i|i\rangle$ and $\langle i|v\rangle = v_i$, then

$$\left(\sum_i |i\rangle\langle i|\right)|v\rangle = \sum_i |i\rangle\langle i|v\rangle = \sum_i v_i|i\rangle = |v\rangle \ .$$

This equality holds for all $|v\rangle \in V$, thus $\sum_i |i\rangle\langle i|$ must be the unit operator and it is known as *completeness relation* for orthonormal vectors.

The spectral decomposition is an extremely useful representation theorem for normal operators.

Definition 4.1.1. *A normal operator on a complex Hilbert space $\mathcal{H}$ is a continuous linear operator $N : \mathcal{H} \Rightarrow \mathcal{H}$ that commutes with its Hermitian adjoint N^**

$$NN^* = N^*N \ .$$

In particular we call N a unitary operator if $N^* = N^{-1}$ and N an hermitian operator if $N^* = N$.

Theorem 4.1.2. *An operator A on a vector space V is normal if and only if it is diagonalizable, i.e. the transformation matrix of this operator is diagonal respect to some orthonormal basis for V.*

4.2 Postulates of the Quantum Mechanics

Now, we are in position to enunciate the four postulates of quantum mechanics: they must be consider as a set of basic statements representing a starting point of quantum theory in axiomatic form.
First postulate. (Quantum states). *We associate each physical system with a Hilbert space $\mathcal{H}$, representing the space of the possible states of the system. Since vectors that differ by a phase factor are physically indistinguishable states, we can identify any state by a unite vector in this Hilbert space $\mathcal{H}$.*

The most simple quantum system that we know is the qubit (the quantum analogue to classical bits), associated to a 2-dimensional Hilbert space, isomorphic to $\mathbb{C}^2$.

In addition to the basis states, $|0\rangle$ and $|1\rangle$, we can have all the other possible states defined by their linear combinations also called linear overlaps:

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle, \quad \text{with} \quad c_0, c_1 \in \mathbb{C} \quad \text{and} \quad |c_0|^2 + |c_1|^2 = 1 \ .$$

Considering $c_j = r_j e^{i\phi_j}$ with $j = 0, 1$ we can remove $e^{i\phi_0}$, phase factor common to all the components (and so it does not lead to observable effects).

$$|\psi\rangle = r_0|0\rangle + r_1 e^{i(\phi_1 - \phi_0)}|1\rangle \ .$$

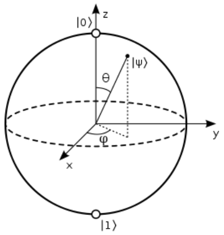
Furthermore, if ψ has unit norm, then

$$|\psi\rangle = \cos\frac{\vartheta}{2}|0\rangle + e^{i\phi}\sin\frac{\vartheta}{2}|1\rangle \ .$$

In that case, the state is parametrizated by the angles ϑ and $\phi = \phi_1 - \phi_0$. They all are spherical coordinates of a point on the surface of a sphere with unit radius S^2.

Using the Cartesian coordinates,

$$x = \cos\phi\sin\vartheta, \quad y = \sin\phi\sin\vartheta \quad z = \cos\vartheta \ .$$



The points of the surface of S^2 parametrize the Hilbert space of the state of a single qubit: the $|0\rangle$ (North Pole) and the $|1\rangle$ (South Pole) are the only points that correspond to classical bits. All other points do not correspond to something classical and represent a non-trivial superposition of basis states.

Unlike the bit, that can only have values 0 and 1, the qubit can be in one of the infinite points on the surface of the Block sphere.

Second postulate. (Composite system). *The space of the states of a composite system is the tensor product of the spaces of states of each subsets.*

Let consider two non-interacting systems A and A'. We associate to them $\mathcal{H}_\mathcal{A}$ and $\mathcal{H}_{\mathcal{A}'}$ respectively. The Hilbert space of the composite system is the tensor product:

$$\mathcal{H}_\mathcal{A} \otimes \mathcal{H}_{\mathcal{A}'} \ .$$

If the first system is in the state $|\psi\rangle_A$ and the second in the state $|\psi'\rangle_{A'}$, the state of the composite system is:

$$|\psi\rangle_A \otimes |\psi'\rangle_{A'} \ .$$

We call this kind of states *separable states*, or product states.

Not all the states are separable: fix a basis $\{|i\rangle_A\}$ for $\mathcal{H}_A$ and a basis $\{|j\rangle_{A'}\}$. The most general state in $\mathcal{H}_\mathcal{A} \otimes \mathcal{H}_{\mathcal{A}'}$ is of the form

$$|\phi\rangle_{AA'} = \sum_{i,j} c_{i,j} |i\rangle_A \otimes |j\rangle_{A'} \ .$$

The state is separable if $c_{i,j} = c_i^A c_j^{A'}$, that is $|\psi\rangle_A = \sum_i c_i^A |i\rangle_A$ and $|\psi'\rangle_{A'} = \sum_j c_j^{A'} |j\rangle_A$

Otherwise, the state is called non-separable or entangled state.

Third postulate.(System evolution). *Every physical process concerning an isolated system is described by an unitary transformation on the space of states.*

We take the states $|\psi\rangle$ at the time t and $|\psi'\rangle$ at the time t'. They are related by an unitary operator

$$U : |\psi'\rangle = U|\psi\rangle \ .$$

The unitary operator in the Hilbert space associated to $\mathbb{C}^{2^n}$ has to be seen as a quantum gates that acts on a set of qubits. Thus, the evolution of an isolated system can be seen as a computational process.

The most important single qubit unitary operators are the $\mathbb{I}$ matrix and the following **Pauli operators**:

$$\sigma_1 = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\sigma_2 = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\sigma_3 = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Forth postulate. (Measure). *The measure of the observable M is described by an Hermitian operator M on the spaces of states. Let $M = \sum_m mP_m$ the spectral decomposition with P_m projectors on the eigenspaces of M. Thus, the possible results of the measure will be the corresponding eigenvalues m.*

When we measure the observable M and the system is in the state $|\psi\rangle$, the probability of obtaining the result m is

$$Pr(m) = \langle\psi|P_m|\psi\rangle \ ,$$

while, the system's state immediately after the measure is

$$|\psi'\rangle = \frac{P_m|\psi\rangle}{\sqrt{p(m)}} \ .$$

Furthermore, the average of the result of the measure is

$$\langle M\rangle \equiv \sum_m mPr(m) = \sum_m m\langle\psi|P_m|\psi\rangle = \langle\psi|M|\psi\rangle \ . \qquad (4.1)$$

4.3 The quantum computer

A quantum computer is a device for the treatment of the information that uses the typical phenomena of quantum mechanics. A classical computer measures the amount of data in bits while the elementary information of a quantum computer, as already mentioned, is the qubit. The principle is that the physical properties of quantum particles can be used to represent data structures, and quantum mechanics can be used to perform operations on these data.

Keeping on the analogy between classical and quantum computer, as a classical computer consists of an electrical circuit with logic gates, a quantum computer consists of a quantic circuit with logic quantum gates, which manipulate the information. One of the simplest classical single-bit gate is the **NOT** gate:

$$0 \rightarrow 1 \quad 1 \rightarrow 0\ .$$

Instead, the quantum NOT-gate acts in the following linear way:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|1\rangle + \beta|0\rangle\ .$$

We can also represent it by the matrix:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Thus, having the quantum state $\alpha|0\rangle + \beta|1\rangle$ written in a vector form $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$

$|0\rangle$ is replaced by the state corresponding to the first column of X, while $|1\rangle$ is replaced by the second column of X. Then,

$$X \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

Any other unitary matrix can be seen as a quantum gate.

Unlink the classical gates, there are many single-quantum bit gates that are not trivial. The most important are:

X-gate $X = \sigma_x$ It exchanges the column vector in a row vector (with the same values).

Y-gate $Y = \sigma_y$ It exchanges the column vector in a row vector and it exchanges the values (multiply the first for $-i$ and the second for i.)

Z-gate $Z = \sigma_z$ It leaves fixed $|0\rangle$ and exchanges the sign of$|1\rangle$;

H-gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{4.2}$$

It sends $|0\rangle$ into $(|0\rangle + |1\rangle)/\sqrt{2}$, and $|1\rangle$ into $(|0\rangle - |1\rangle)/\sqrt{2}$.

The most important example of 2$-$qubit quantum gate is the **CNOT**-gate. It acts on two qubits: the first one is the control qubit $|c\rangle$ and the second one is the target qubits $|t\rangle$. It flips the target if and only if the control qubit is 1.

In this case the transformation U works on a target qubit conditioned by the control qubit. Namely, $U^c = U$ for $c = 1$ and $U^c = \mathbb{I}$ if $c = 0$. Thus, the **CNOT**-gate is the quantum analogue for **XOR**-gate:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Definition 4.3.1. *A set of universal quantum gates is any set of gates to which any possible operation on a quantum computer can be reduced. That is, any other unitary operation on a finite dimensional state space can be expressed as a finite sequence of gates from the set.*

Theorem 4.3.2. *The* 1*-qubit gates together with the **CNOT**-gate form an universal set, i.e.*

$$U \in U(2) \cup \{\textbf{CNOT}\}$$

We refer to [7] for the proof.

We end this section by giving the definition of the *Hadamard test*, that will be used in the algorithm that we will describe in Chapter 5.

Definition 4.3.3. *The Hadamard test acts in this way:*
we start with a two-register state

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\alpha\rangle$$

and we apply Q conditioned on the first qubit to get the state

$$\frac{1}{\sqrt{2}}([|0\rangle \otimes |\alpha\rangle] + |1\rangle \otimes Q|\alpha\rangle) \ ,$$

then we apply the Hadamard gate (4.2) on the first qubit and we output the measurement. The output is 1 *if the measurement result is $|1\rangle$, while is -1 if the measurement result is $|0\rangle$. To get the random variable with the expectation value is the imaginary part we have to start with the state $\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \otimes |\alpha\rangle$.*

Property 4.3.4. *If a state $|\alpha\rangle$ can be generated and a unitary matrix Q can be applied, both efficiently, then there exists an efficient quantum circuit whose output is a random variable $\in \{1, -1\}$ and whose expectation value is $\mathcal{R}e\langle\alpha|Q|\alpha\rangle$ (or $\mathcal{I}m\langle\alpha|Q|\alpha\rangle$.)*

Proof. The proof follows from the fourth postulate of the Quantum Mechanic. In particular, from the equality that describe the average of the result of the measure of an observable (4.1). □

4.4 Classes of computational complexity.

The notion of quantum Turing machine was introduced by Deutsch as a quantum version of the classical Turing machine Tm.

The quantum Turing machine is the theoretical model of a quantum computer, and it is known as the universal quantum computer.

Definition 4.4.1. *A quantum Turing machine (qTm) is a machine with a finite number of states that has the following three fundamental components.*

1. *A finite process. It consists of a finite number p of qubits. We denote the Hilbert space of the states of the process $\mathcal{H}_P$ with the basis $\{\otimes_i |p_i\rangle : p_i = 0, 1\}_{i=0}^{p-1}$;*

2. *A memory tape. It consists of a infinite number of qubits. Ideally there is a qubit per cell and only a finite number of them will be active in each computational step. We denote the Hilbert space of the states of the process $\mathcal{H}_M$ with the basis $\{\otimes_i |m_i\rangle : m_i = 0, 1\}_{i=-\infty}^{+\infty}$;*

3. *A cursor. It represents the interacting components between the control units and memory tape. Its position is given by the variable $x \in \mathbb{Z}$ and the Hilbert state associated is $\mathcal{H}_C$ with the basis $\{|x\rangle : x \in \mathbb{Z}\}$*

Overall, le Hilbert space that describe the space of the state of the quantistic Turing machine is

$$\mathcal{H}_{qTM} = \mathcal{H}_C \otimes \mathcal{H}_P \otimes \mathcal{H}_M \ .$$

The basis vectors are

$$|x; \mathbf{p}; \mathbf{m}\rangle = |x; p_0, p_1, \ldots, p_{P_1}; \ldots, m_{-1}, m_0, m_{+1}, \ldots\rangle$$

and they represent the states of the computational basis.

A quantum Turing machine works in fixed T-period steps and during each step only the process and a finite part of the memory interact through the cursor. While in the classical Turing machine we have a set of instructions, here we have an unitary time evolution of the quantum states $|\psi\rangle \in \mathcal{H}_{QTM}$. For example after n computational steps, the quantum Turing machine state will be:

$$|\psi(nT)\rangle = U^n |\psi(0)\rangle \text{ with } U \text{ unitary operator.}$$

Now we are going to describe the computation of this machine. At each computational step the framework x is examined and the state m_x of the framework x itself is read.

If the inner state is $\mathbf{p}$, then the framework x is put in the state m'_x. After that, it is reached the inner state **p'** and the head is shifted to one step to the left, or to the right, or to both the direction.

Starting to a basis vector we get, after some computational steps, a state that is the coherent superposition between different inner states of the machine, different position of the cursor and different states of the memory tape (*entangled state*).

Once finished a convergent computation, the output is read by making a measure on the tape itself. This process projects the state of the tape on one of the possible results. Each of them is obtained with a probability depending on the transition amplitudes of the computational steps.

Chapter 5

The algorithm

5.1 Classical algorithms

We have four classical algorithms that compute the Jones polynomial. They take an oriented knot K as the input and return the Jones polynomial $V_K(x)$ as the output. More precisely, the input is given by an oriented diagram D of K and its complexity is the number $n = n(D)$ of the crossings occurring in D.

The first algorithm is based on the Kauffman approach to the Jones polynomial, that is on definitions (1.3.5) and (1.3.9).

```
KAUFFMAN1 ($K$: oriented knot represented by an oriented diagram $D$)

begin

    compute $w(D)$ ;
    $\zeta(D) := 0$ ;
    for any states $s$ of $D$
        compute $\sigma(s)$ and $\gamma(s)$ ;
        $\vartheta(s) = t^{\sigma(s)}(-t^{-2} - t^{2})^{\gamma(s)-1}$ ;
        $\zeta(D) = \zeta(D) + \vartheta(s)$ ;
    $P_K(t) := t^{-3w(D)}\zeta(D)$ ;
    $V_K(x) := P_K(x^{-\frac{1}{4}})$ ;
```

```
        return $V_K(x)$ ;
    end.
```

Notice that the computation of $w(D)$ only require a linear time with respect to n. The reason why the algorithm takes an exponential time is that we have to itemize all the 2^n states of the diagram D.

The second algorithm, due to Conway, is based on the skein relation (1.1).

```
CONWAY ($K$: oriented knot represented by an oriented diagram $D$)
begin
    if $K$ is the unknot
    then $C(K) = 1$ ;
    else
        choose one crossings of $D$ to be inverted for making $K$ into unknot;
        set $K_\pm = K$ depending on the sign of the choosen crossing ;
        set $K_\mp =$ the knot obtained from $K$ by inverting the choosen crossings ;
        set $K_0 =$ the knot obtained from $K$ by resolving the choosen crossing ;
        $V_{K_\mp}(x) = \text{CONWAY}(K_\mp)$ ;
        $V_{K_0}(x) = \text{CONWAY}(K_0)$ ;
        $V_{K_\pm}(x) = x^{\mp 1}(x^{\mp 1} V_{K_\mp}(x) \pm (x^{-1/2} - x^{1/2}) V_{K_0}(x))$ ;
        return $V_{K_\pm}(x)$ ;
end
```

Actually, this algorithm can only be used for knots that have few crossings. In fact, determining if k is an unknot is not a trivial action and it does not take a polynomial time of resolution. The algorithm can be reformulated, fixing a priori a sequence of crossings that we have to inverted in order to reduce K to the unknot. In this case we proceed in an iterative manner and we do not have to check if the knot is trivial. The choice of the sequence of the crossings mentioned before is made in a linear way.

The complexity of thus algorithm is exponential because the algorithm referes twice to itself.

The third algorithm is based on the genuine definition thought by Jones himself. It is based on theorem (2.4.4). In this algorithm we refere to another algorithm that computes the Tr_n function.

```
$\mathrm{TRACE}_n(h = t_{i_1}^{\varepsilon_1} t_{i_2}^{\varepsilon_2} \cdots t_{i_{\gamma(b)}}^{\varepsilon_{\gamma(b)}})$
begin
    if $h = 1$
    then $\mathrm{Tr}_n(h)$ := 1 ;
    else
        put $h$ in the form$h = a + \sum_i x_i t_{n-1} y_i$, where $a, x_i, y_i \in \mathbb{H}_{n-1}$
        $Tr_n(h) = \mathrm{TRACE}_{n-1}(a) + z \sum_i \mathrm{TRACE}_{n-1}(x_i y_i)$
return $Tr_n(h)$
end
```

We observe that the algorithm ends when we reach $n = 1$ and the value $\mathrm{Tr}_1(h) = \mathrm{Tr}_n(1) = 1$, because the braid has no crossings.

Using tha above algorithm we introduce the Jones algorithm.

```
JONES ($K$: oriented knot)
begin
    $b$ := braid such that $K = b^{tr}$ with $n(b)$ strands and $\gamma(b)$ crossings ;
    set $n(b)$ ;
    set $\gamma(b)$ ;
    $c$ := $\left(\frac{1}{z}\right)^{(n(b)+\gamma(b)-1)/2} \left(\frac{q}{z-q+1}\right)^{(n(b)-\gamma(b)-1)/2}$ ;
    $h$ := $\rho(b) = t_{i_1}^{\varepsilon_1} t_{i_2}^{\varepsilon_2} \cdots t_{i_{\gamma(b)}}^{\varepsilon_{\gamma(b)}}$ ;
    $W_K(q, z) = c\ \mathrm{TRACE}_n(h)$;
    $P_K(x, y) = W_K((\frac{z}{z-q+})^{\frac{1}{2}}, q^{\frac{1}{2}} - q^{-\frac{1}{2}})$;
```

```
        $V_K(x) = (x, x^{\frac{1}{2}} - x^{-\frac{1}{2}})$;
        return $V_K(x)$
end
```

This algorithm take an exponential time of resolution because the subprocedure that computes Tr_n referes twice to itself. In order to write the knot K in term of braid closure (b^{tr}) we refer to the Vogel algorithm [11]. This algorithm can be performed in a order of n^2 time complexity, with n number of strands of b^{tr}.

The fourth algorithm uses the TL_n algebras. It is based on the representation on the B_n group into $TL_n(d)$, in particular we refer to the definition (3.3.2).

```
KAUFFMAN2 ($K$: oriented knot)
begin
$b$ := braid obtained after the trace closure of $K$ ;
compute $w(K)$;
$g := \rho_d(K) = (ae_{i_1}^{\varepsilon_1} + a^{-1}1)(ae_{i_2}^{\varepsilon_2} + a^{-1}1) \cdots (ae_{i_n}^{\varepsilon_n} + a^{-1}1)$;
expand $g$ as the sum of monomial terms $t_1 + t_2 + \cdots + t_{2^n}$
for any $t_i$
        compute $\gamma(t_i)$, where $\gamma(t_i)$ is the number of loops in the corresponding
            diagram of $t_i$;
        $tr(t_i) = d^{\gamma(t_i)-n}$ ;
$tr(K) = \sum_i tr(t_i)$ (that is the sum on all the elements that occur in $g$);
$P_K(t) = (-t)^{-3w(k)}(-t^{-2} - t^2)^{n-1} tr(g)$;
$V_K(x) = P_K(x^{-\frac{1}{4}})$;
return $V_K(x)$
end
```

This algorithm takes an exponential time of resolution due to the expansion of $\rho_d(K)$ in 2^n factors.

It is important to underline the fact that all the above algorithms calculate the exact Jones polynomial with a computational time that is exponential with respect to the complexity of the input.

In principle, they take the same exponential time even when used to evaluate the Jones polynomial at a single complex number.

5.2 Implementing the path model representation

Here we want to show an explicit and simple quantum algorithm, both for the trace closure b^{tr} and for the plat closure b^{pl} of a braid b [8], that approximate the value of the Jones polynomial at the single complex number $t = e^{\frac{2\pi i}{k}}$. This algorithm is efficient and can be performed in a polynomial time. However, we do not know yet how this algorithm can approximate the Jones polynomial and we refer to section (5.5).

First, we adapt the path model representation (section 3.4) in order to work on qubits.

Definition 5.2.1. *We define $P_{n,k,\ell}$ to be the set of all sequences of bits $(p_1, p_2, \ldots, p_n)$ describing a path $q(0), q(1), \ldots, q(n)$ in $Q_{n,k,\ell}$ (see section 3.4) according to the following rule:*

$$p_i = \begin{cases} 0 & \text{if } q(i) = q(i-1) - 1 \\ 1 & \text{if } q(i) = q(i-1) + 1 \end{cases} \tag{5.1}$$

Likewise, we define $P_{n,k}$ to be the set of all sequences describing paths of length n with no restriction on the final point. Thus, $P_{n,k} = \bigcup_{\ell=1}^{k} P_{n,k,\ell}$.

We define $\mathcal{H}_{n,k,\ell}$ as the span of $P_{n,k,\ell}$ in the Hilbert space of n qubits and $\mathcal{H}_{n,k}$ as the span of $P_{n,k}$.

Moreover, we denote by

$$\eta : \mathcal{V}_{n,k} \to \mathcal{H}_{n,k} \tag{5.2}$$

the isomorphism sending the base element $(q(0), q(1), \ldots, q(n))$ of $\mathcal{V}_{n,k}$ to the corresponding base element $(p_1, p_2, \ldots, p_n)$ of $\mathcal{H}_{n,k}$

According to this isomorphism, we can define another linear representation of $TL_n(d)$, by composing the linear representation $\tau : TL_n(d) \to End(\mathcal{V}_{n,k})$, already defined in section (3.2) and the $\eta_* : End(\mathcal{V}_{n,k}) \to End(\mathcal{H}_{n,k})$, isomorphism induced by the isomorphism η, i.e.

$$\Phi : TL_n(d) \to End(\mathcal{H}_{n,k}) \ .$$

In order to see how Φ acts, let us apply it to each basis element $e_i \in TL_n(d)$ and denote by $\Phi_i = \Phi(e_i)$ its image.

Let $p = (p_1, p_2, \ldots, p_n)$ be the sequence of bit describing the path $(q(0), q(1), \ldots, q(n))$ in G_k. We have these formulas that define the action of Φ_i:

$$\Phi_i|p_1, \ldots, p_{i-1}, 0, 0, p_{i+2}, \ldots, p_n\rangle = \quad 0 \tag{5.3}$$

$$\Phi_i|p_1, \ldots, p_{i-1}, 0, 1, p_{i+2}, \ldots, p_n\rangle = \tag{5.4}$$

$$= \frac{\lambda_{q(i-1)}}{\lambda_{q(i)}}|p_1, \ldots, p_{i-1}, 0, 1, p_{i+1}, \ldots, p_n\rangle + \frac{\sqrt{\lambda_{q(i+1)}\lambda_{q(i-1)}}}{\lambda_{q(i)}}|p_1, \ldots, p_{i-1}, 1, 0, p_{i+1}, \ldots, p_n\rangle$$

$$\Phi_i|p_1, \ldots, p_{i-1}, 1, 0, p_{i+1}, \ldots, p_n\rangle = \tag{5.5}$$

$$\frac{\lambda_{q(i+1)}}{\lambda_{q(i)}}|p_1, \ldots, p_{i-1}, 1, 0, p_{i+1}, \ldots, p_n\rangle + \frac{\sqrt{\lambda_{q(i+1)}\lambda_{q(i-1)}}}{\lambda_{q(i)}}|p_1, \ldots, p_{i-1}, 0, 1, p_{i+1}, \ldots, p_n\rangle$$

$$\Phi_i|p_1, \ldots, p_{i-1}, 1, 1, p_{i+1}, \ldots, p_n\rangle = \quad 0 \tag{5.6}$$

We remember, from the 3.4.3, that $\lambda_\ell = \sin\left(\frac{\pi\ell}{k}\right)$ and in the above formulas (5.3-5.6) we adopt the convention that $\lambda_j = 0 \ \forall j \notin \{1, \ldots, k-1\}$.

Proposition 5.2.2. *Φ is a unitary representation of $TL_n(d)$ and Φ_i are Hermitian. Moreover, Φ induces a unitary representation $\phi : B_n \to End(\mathcal{H}_{n,k})$ by componing with ρ_d defined in (3.1). of the braid group B_n, operating on $\mathcal{H}_{n,k}$.*

Proof. The proof follows from the properties of τ and the natural isomorphism describe in definition (5.2). □

The operators $\phi_i = \phi(b_i)$ on $\mathcal{H}_{n,k}$ can be extended to the rest of the sequences of $p_i \notin P_{n,k}$.

Proposition 5.2.3. *For all $i \in \{1, \ldots, n\}$, ϕ_i can be implemented on the entire Hilbert space $\mathcal{H}$ of n qubits using poly(n, k) gates.*

Proof. See [8]. □

Thus, we deduce that

Corollary 5.2.4. *For every $b \in B_n$, with m crossings, there exists a quantum circuit* Q(b) *that applies $\phi(b)$ to n qubits, using poly(m, n, k) elementary gates.*

Definition 5.2.5. *We define* $\mathrm{Tr}_{\Phi,n}(W)$ *for all the $W \in \Phi(TL_n(d))$ as*

$$\mathrm{Tr}_{\Phi,n}(W) = \frac{1}{N} \sum_{\ell=1}^{k-1} \lambda_\ell \, \mathrm{Tr}(W|_\ell) \ ,$$

where $W|_\ell$ is the restriction of W in the subspace $\mathcal{H}_{n,k,\ell}$, Tr *is the standard trace of the matrices and $N = \sum_\ell \lambda_\ell \dim(\mathcal{H}_{n,\ell,k})$ (this sum is taken over all the ℓ such that $P_{n,k,\ell}$ is not empty.)*

We observe that:

1. for any $t \in TL_n(d)$, $\Phi(t)(\mathcal{H}_{n,k,\ell}) \subseteq \mathcal{H}_{n,k,\ell}$: in fact Φ_i cannot change the final point of a path since it only moves 01 to 10 and vice versa;

2. we claim that the function $\mathrm{Tr}_{\Phi,n}(\cdot)$ is a Markov trace, that is it satisfies the property in (3.3.1);

Thus, by the uniqueness of the Markov trace, we have that $\mathrm{Tr}_{\Phi,n}(\phi(b)) = tr(\rho_d(b))$ and, using the formula (3.5.1), we have

Lemma 5.2.6. *For all $b \in B_n$,*

$$V_{b^{tr}}(a^{-4}) = P_{b^{tr}}(a) = (-a)^{3w(B^{tr})} d^{n-1} \, \mathrm{Tr}_{\Phi,n}(\phi(b)) \ .$$

5.3 The Algorithm Approximate-Jones-Trace-Closure

We describe the algorithm already studied in [8] for a trace closure b^{tr} of a braid b and we call it **AJTC**.

```
AJTC($K = b^{tr}$, $a = e^{-\frac{2\pi i}{k}}$);
begin
d := $-a^2 - a^{-2} = 2\cos(\frac{\pi}{k})$ ;
compute $w(b^{tr})$ ;
for j=1:poly(m,n,k)
    pick the state corresponding to a random path $p \in P_{n,k}$ such
        that $Pr(p) \propto \lambda_\ell$, with $\ell = q(n)$ ;
    apply the Hadamard test to the operator $\phi(b)$ and the state $p$;
    compute $x_j$ ;
    compute $y_j$ ;
$r$ := average $(x_j + y_j)$ ;
$O_{tr}$ $= (-a)^{3w(B^{tr})} d^{n-1} r$;
return $O_{tr}$ .
```

Lemma 5.3.1. *There exists a classical probabilistic algorithm that outputs a random path $P_{n,k,\ell}$ according to a distribution close to the uniform and this algorithm works in polynomial time.*

Theorem 5.3.2. *Given a braid $b \in B_n$ with n strands and m crossings and given $k \in \mathbb{Z}$, in polynomial time poly(m, n, k) the quantum algorithm **AJTC**, except for an exponentially small probability, outputs $O_{tr} \in \mathbb{C}$ such that*

$$|O_{tr} - V_{B^{tr}}(e^{\frac{2\pi i}{k}})| < \varepsilon \left(2\cos\left(\frac{\pi}{k}\right)\right)^{n-1}$$

with $d = -a^2 - a^{-2} = 2\cos(\frac{\pi}{k})$ and $\varepsilon = \frac{1}{poly(m,n,k)}$.

Proof. Due to the lemma (5.2.6) the correctness of the Algorithm Approximate-Jones-Trace-Closure follows simply because the output r satisfies

$$|r - \mathrm{Tr}_{\Phi,n}(\phi(b))| \leq \varepsilon$$

with $\varepsilon = \frac{1}{poly(m,n,k)}$ time, except for an exponentially small probability.

The Hadamard test implies that the expectation value of x_j for a fixed p is $\mathcal{R}e(\langle p|\phi(b)|p\rangle)$.

The same expectation taken over a random p is

$$\frac{\sum_{\ell,p\in P_{n,k,\ell}} \lambda_\ell \mathcal{R}e(\langle p|\phi(b)|p\rangle)}{\sum_{\ell,p\in P_{n,k,\ell}} \lambda_\ell} = \frac{\sum_\ell \lambda_\ell \mathcal{R}e(\mathrm{Tr}(\phi(b)|_\ell))}{\sum_\ell \lambda_\ell \dim(\mathcal{H}_{n,k,\ell})} = \mathcal{R}e(\mathrm{Tr}_{\Phi,n}(\phi(b))) \ .$$

The same work has to be done for the imaginary part.

We consider the value x_j with the Bernoulli distribution: it takes the value 1 with probability p and -1 with probability $1-p$. We will call the value x_j as ξ.

Its expected values is

$$E(\xi) = -1\cdot(1-p) + 1\cdot p = 2p-1 = \mu$$

Its variance is

$$D(\xi) = E(\xi^2) - (E(\xi))^2 = 1-(2p-1)^2 =$$
$$= 1-(4p^2-4p+1) = 4p-4p^2 = 4p(1-p) = \sigma^2$$

Now we consider the independent sum of random variables all distributed with the Bernoulli distribution: $\eta_n = \frac{1}{n}\sum_{i=1}^n \xi_i$. We have that $E(\eta_n) = \mu$ and $D(\eta_n) = \frac{\sigma^2}{n}$.

The normalized random variable is $\overline{\eta_n} = \frac{\eta_n - \mu}{\frac{\sigma}{\sqrt{n}}}$.

Thus, the probability that this normalized random variable is less then a value x goes asymptotically as $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}dt$ and we can found the estimation of $P(|\eta_n - \mu| > \varepsilon)$ that is equal to $P(|\overline{\eta_n}| > \frac{\varepsilon\sqrt{n}}{\sigma})$. It goes asymptotically as $\frac{\sigma}{\varepsilon\sqrt{n}}e^{\frac{\varepsilon^2 n}{2\sigma^2}}$.(Feller, An Introduction on probability theory and its applications)

Thus, the output r satisfies the inequality $|r - \mathrm{Tr}_n(\phi(b))| \leq \varepsilon$ except for an exponentially small probability.

The proof of the Theorem (5.3.2) follows. □

5.4 The Algorithm Approximate-Jones-Plat-Closure

We describe the algorithm already studied in [8] for a plat closure b^{pl} of a braid b and we call it **AJPC**.

```
AJPC($K = b^{pl}$, $a = e^{-\frac{2\pi i}{k}}$);
begin
$\lambda_\ell$  :=  $\sin\frac{\pi\ell}{k}$  ;
N    :=  $\sum_\ell \lambda_\ell \dim(\mathcal{H}_{n,k,\ell})$  ;
compute $w(b^{pl})$ ;
generate the state $|\alpha\rangle = |1,0,1,0,\ldots,1,0\rangle$ ;
for j=1:poly(m,n,k)
    apply the Hadamard test to the operator $\phi(b)$ and to $|\alpha\rangle$ ;
    compute $x_j$ ;
    compute $y_j$ ;
compute $r$  := average ($x_j + y_j$) ;
$O_{pl}$  :=  $(-a)^{3w(B^{pl})} d^{\frac{3n}{2}-1} \lambda_1 \frac{r}{N}$;
return $O_{pl}$ ;
```

Theorem 5.4.1. *Given a $b \in B_n$ with n strands and m crossings and given $k \in \mathbb{Z}$, in polynomial time poly(n, m, k) the quantum algorithm **AJPC** and except a exponentially small probability, outputs $O_{pl} \in \mathbb{C}$ with*

$$|O_{pl} - V_{B^{pl}}(e^{\frac{2\pi i}{k}})| < \frac{\varepsilon(2\cos(\frac{\pi}{k}))^{\frac{3n}{2}}}{N}$$

with $d = -a^2 - a^{-2} = 2\cos(\frac{\pi}{k})$, ε the inverse of a polynomial in m,n and k and $N = \sum_\ell \lambda_\ell \dim(\mathcal{H}_{n,k,\ell})$ an exponentially big factor.

Proof. We observe that the plat closure of a braid b is isotopic to the trace closure of a tangle c, that is a braid in which some of its crossings have been replaced by a picture of a capcups form (see figure in definition 3.5).

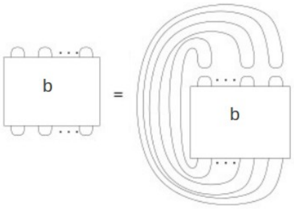


Figure 5.1: $c = b \cdot \frac{n}{2}$ capcups.

Thus, we relate the Jones polynomial of c^{tr} to $\mathrm{Tr}(\phi(b)|\alpha\rangle\langle\alpha|)$. The thesis is now written in terms of trace closure a traces functions and we can apply the formula (3.5.1)

The value of r can be approximated with $\langle\alpha|\phi(b)|\alpha\rangle$, which is equal to $\mathrm{Tr}(\phi(b)|\alpha\rangle\langle\alpha|)$.

Finally, we need to describe better the connection between the projection on $|\alpha\rangle$ and the capcups.

It is easy to verify that $\Phi_1\Phi_3\ldots\Phi_{n-1}$ applied to any other path different from $|\alpha\rangle$ gives 0. In fact, according to the definition (5.3-5.6) we know that the operator Φ_i gives a non zero output only in two cases: (5.4) and (5.5).

Moreover, we restricted the strings to those that describe a path starting to the leftmost vertex of G_k and thus the path never can start with a 0 which means to take a step to the left (according to the definition), thus the only possible path will be $|1,0,1,0,\ldots,1,0\rangle$ that is exactly $|\alpha\rangle$.

The Φ_i commutes if their indices are more than one apart: so we can apply Φ_1, then Φ_3 and so on. We also remember that the Φ_i operate on a single pair of coordinates (the first two, then the second two, and so on):

$$\Phi_{n-1}\Phi_{n-3}\ldots\Phi_3\Phi_1|\underline{1,0},1,0,\ldots,1,0\rangle =$$

$$= \Phi_{n-1}\Phi_{n-3}\ldots\Phi_3\left(\frac{\lambda_2}{\lambda_1}|1,0,1,0,\ldots,1,0\rangle + \frac{\sqrt{\lambda_2\lambda_0}}{\lambda_1}|1,0,1,0,\ldots,1,0\rangle\right) =$$

$$= \Phi_{n-1}\Phi_{n-3}\ldots\Phi_3\left(\frac{\lambda_2}{\lambda_1}|1,0,1,0,\ldots,1,0\rangle\right)$$

$$\ldots$$

$$\left(\frac{\lambda_2}{\lambda_1}\right)^{\frac{n}{2}}|1,0,1,0,\ldots,1,0\rangle = \left(\frac{\lambda_2}{\lambda_1}\right)^{\frac{n}{2}}|\alpha\rangle = d^{\frac{n}{2}}|\alpha\rangle \ .$$

Thus,

$$\frac{\Phi_{n-1}\Phi_{n-3}\ldots\Phi_3}{d^{\frac{n}{2}}} = |\alpha\rangle\langle\alpha| \ . \tag{5.7}$$

Using the definition 5.2.5 and the equation (5.7):

$$\langle\alpha|\phi(b)|\alpha\rangle = \mathrm{Tr}(\phi(b)|\alpha\rangle\langle\alpha|) = \frac{N}{\lambda_1}\,\mathrm{Tr}_n(\phi(b)|\alpha\rangle\langle\alpha|)$$

$$\frac{N}{\lambda_1}\,\mathrm{Tr}_n\left(\phi(b)\frac{\Phi_1\Phi_2\ldots\Phi_{n-1}}{d^{\frac{n}{2}}}\right) = \frac{N}{\lambda_1 d^{\frac{n}{2}}}\,\mathrm{Tr}_n(\phi(C))$$

By the uniqueness of the Markov trace, we have that $\mathrm{Tr}_n(\phi(C)) = tr(\rho_d(C))$. Using 3.5.1:

$$V_{b^{pl}}(a^{-4}) = V_{c^{tr}}(a^{-4}) = P_{c^{tr}}(a) = (-a)^{3w(c^{tr})}d^{n-1}\,\mathrm{Tr}_n(\phi(c))$$

Thus, the proof of the Theorem is complete. □

5.5 Conclusion e further direction

We want to mention the result of Aharonov and Arad concerning the complexity of the problem of approximating the values of the Jones polynomial [12].

Theorem 5.5.1. *The problem of approximating the Jones polynomial of the plate closure of a braid b at* $e^{\frac{2\pi i}{k}}$*, for k polynomial with respect to the size of the input, with the accuracy given by Theorem 5.4.1, is BQP-hard.*

BQP stands for *bounded-error quantum polynomial.* This is the class of decision problems solvable by a quantum Turing machine in polynomial time with error probability of at most $\frac{1}{3}$ for all instances. It is the quantum analogue of the completely class BPP (bounded-error probabilistic polynomial). This is the class of decision problems solvable by a probabilistic Turing machine in polynomial time, with an error probability of at most 1/3 for all instances.

A BQP-hard problem is a problem *at least as hard as the hardest problem in* BQP. More precisely, a problem A is BQP-hard if and only if there is a BQP-complete problem B that is polynomial time Turing-reducible to A.

A BQP-complete problem is the hardest problem in BQP. A is BQP-complete if A is BQP and all the B in BQP are such that $B \leq A$.

Thus, we can hope that this algorithm can be used to construct other algorithms for BQP problems.

Now, we want to briefly discuss the relation between the exact computation of the Jones polynomial and the approximation of its value at the roots of unit $e^{\frac{2\pi i}{k}}$.

We recall that a polynomial $p(z)$, with coefficients in $\mathbb{Q}$ and degree d, is unically determined by its value $p(e^{\frac{2\pi i}{k}})$ at any single root of unit such that $\varphi(n) > d$, where $\varphi(n)$ is the Euler function. The reason is that the minimal polynomial of $e^{\frac{2\pi i}{k}}$ over $\mathbb{Q}$ has degree $\varphi(n)$ [13].

We notice that $P_K(x)$ is a Laurent polynomial with coefficients in $\mathbb{Z}$ and minimal exponent greater $-2n$ [1]. Thus, the polynomial $U_K(x) = V_K(x)x^{2n}$ is a genuine polynomial with coefficient in $\mathbb{Z}$ and degree $d < 3n$ [1]. Moreover, the coefficients of such polynomial belong to the interval $(-2^n, 2^n)$ [15]. Thus, is sufficient to know the exact value of $V_K(e^{\frac{2\pi i}{k}})$ with $\varphi(k) > 3n$. For example, if k prime, we have $k > 3n$ and we know, from the Bertrand's postulate [14], that this such prime exists and it is minor than $6n$.

Now the problem is that the quantum algorithm does not outputs the exact value of the Jones polynomial at $e^{\frac{2\pi i}{k}}$, but just an approximation. In order to know to determine the exact value of $V_K(x)$ it is sufficient to know the minimum distance between any two linear combination of the form

$$a_0 + a_1 e^{\frac{2\pi i}{n}} + a_2 e^{2\frac{2\pi i}{n}} + \cdots + a_d e^{d\frac{2\pi i}{n}} = \sum_j^d a_j e^{j\frac{2\pi i}{n}} \ ,$$

where $3n < k < 6n$ and $a_j \in \mathbb{Z}$, $a_j \in (-2^n, 2^n)$.

Actually, taking into account

$$|\sum_j^d a_j e^{j\frac{2\pi i}{n}} - \sum_j^d a'_j e^{j\frac{2\pi i}{n}}| = |\sum_j^d (a_j - a'_j) e^{j\frac{2\pi i}{n}}| = |\sum_j^d b_j e^{j\frac{2\pi i}{n}}| \ , \tag{5.8}$$

such minimal distance can be reduced to the minimum value of $|\sum_j^d b_j e^{j\frac{2\pi i}{n}}|$ with $b_j \in (-2^{n+1}, 2^{n+1})$.

Now, there are some examples of the values of $V(e^{\frac{2\pi i}{k}})$.

If $n = 2, 3, 4, 6$ we observe in the figures that we have dicrete lattices. The minimum diatance between any two possible values of such polynomial is always 1.

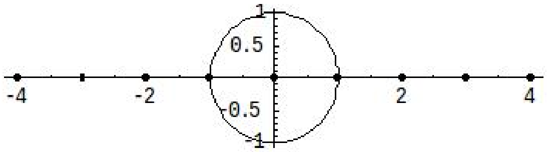


Figure 5.2: $n = 2$, $d = 1$, $h = 2$.

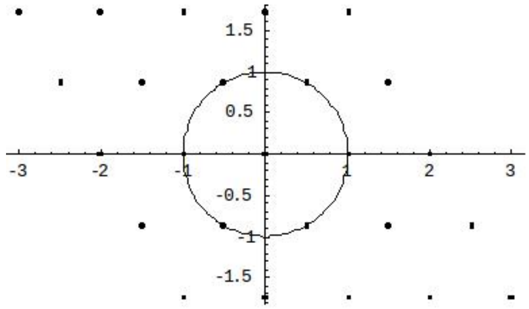


Figure 5.3: $n = 3$, $d = 1$, $h = 2$.

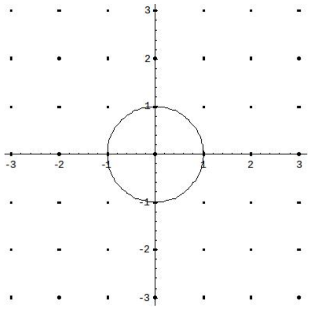


Figure 5.4: $n = 4$, $d = 1$, $h = 3$.

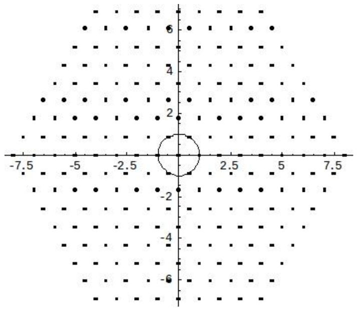


Figure 5.5: $n = 6$, $d = 5$, $h = 2$.

On the contrary, for $n = 5, 7, 8, \ldots$ (and for all the other n), we observe that the more h increase, the more the distance becomes small.

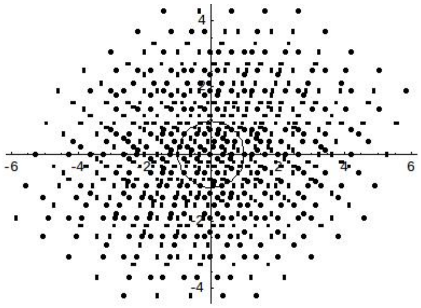


Figure 5.6: $n = 5$, $d = 3$, $h = 2$.

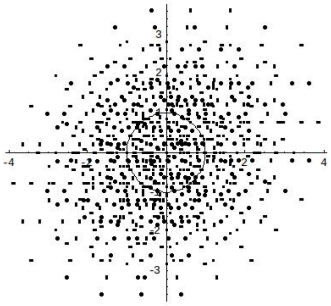


Figure 5.7: $n = 7$, $d = 1$, $h = 5$.

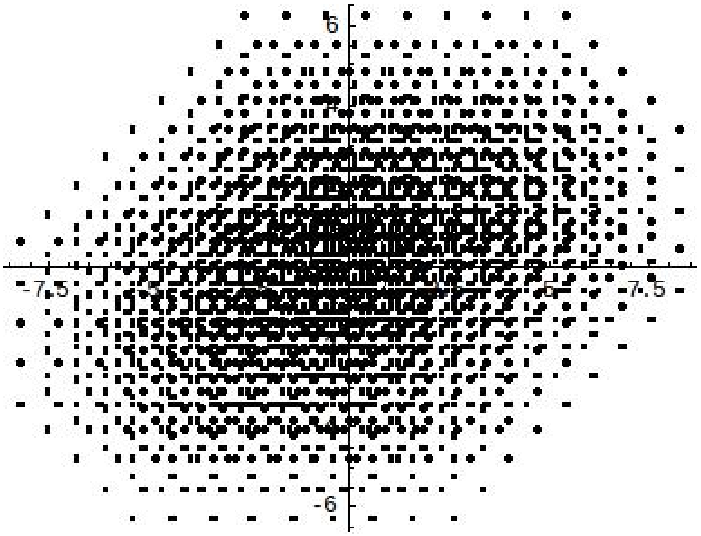


Figure 5.8: $n = 8$, $d = 2$, $h = 5$.

It would be interesting to have an estimate of how this minimal distance can decrease with respect to all the parameters.

Bibliography

[1] W.B.R. Lickorish, *An Introduction to Knot Theory*, Springer 1997.

[2] A. Sossinsky, *Knots, mathematics with a twist*, Harvard University Press, Cambridge-Massachusetts London-England, 2002.

[3] K. Reidemeister, *Elementare Begründung der Knotentheorie*, Abh. Math. Sem. Univ. Hamburg 5, 24-32, 1926.

[4] K. Reidemeister, *Knot theory (Translation of Knotentheorie)*, BSC Associates Moscow, Idaho, 1983.

[5] P. de la Harpe, M. Kervaire, C. Weber, *On the Jones polynomial*, L'Enseign. Math. 32 (1986), 271-335.

[6] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge Univ. Press 2000.

[7] C. Toffalori, F. Corradini, S. Leonesi, S. Mancini, *Teoria della computabilità e della complessità*, Mc-Graw-Hill, 2005.

[8] D. Aharonov, V. Jones, Z. Landau, *A Polynomial Quantum Algorithm for Approximating the Jones Polynomial*, in the Proceedings of the 38th annual ACM Symposium of Theory of Computing, ACM New York 2006.

[9] C. Kassel, V. Turaev, *Braid Groups*, Springer 2008.

[10] D. Bisch, V. Jones, *Algebras associated to intermediate subfactors*, Inventiones Mathematicae, Springer-Verlag, 1997.

[11] V.V. Prasolov, A.B. Sossinsky, *Knots, Links, Braids and* 3*-Manifolds*, American mathematical Society, 2000.

[12] D. Aharonov, I. Arad, *On the BQP-hardness of Approximating the Jones Polynomial*, preprint, 2006.

[13] I.N. Herstein, *Topics in algebra*, Xerox College Publishing, 1975.

[14] G.H. Hardy, E.M. Wright, *An introduction to the theory of numbers*, Oxford University Press, 1975.

[15] A. Stoimenow, *On the coefficients of the link polynomials*, Springer-Verlag, 2003.