HS2010 Symmetries in Quantum Information Theory and Quantum Computation

Matthias Christandl

December 19, 2010

Abstract

This course gives an introduction to Quantum Information Theory and Quantum Computation through the study of symmetries of physical systems. After an introduction to the concept of quantum information as the spin degree of a particle, the course develops this concept in two ways: First, the distillation of entanglement, one of the most fundamental tasks in quantum information theory, is explained as a measurement of the total spin of a bunch of particles. Second, computation is introduced as an exchange of particles, leading to the topological model of quantum computation.

This course aims at an understanding of quantum information as a natural physical concept. On the technical level, methods familiar from a basic course in quantum mechanics will be adapted to studying quantum information theory and computation.

The course is complementary to the courses on quantum information theory by Professor Renato Renner (D-PHYS) and on quantum computation by Professor Stefan Wolf (D-INF). The course is aimed at master students in physics. As prerequisites is a basic course in quantum mechanics. The course language is English.

1 Introduction

1.1 Remarks

Rolf Landauer famously observed that *information is physical* and that every act of *information processing* is a *physical process.* Since the information processing of our interest is done on our planet and with small devices, we are interested in studying the information processing within non-relativistic quantum theory. Quantum Information Science is the name for this subject which is formed at the interface of Computer Science and Quantum Physics.

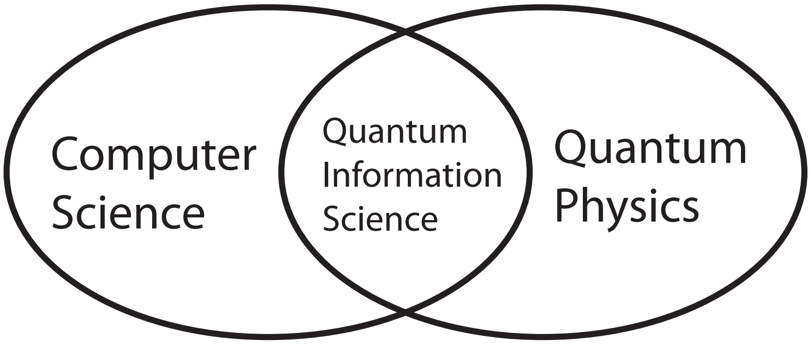


Figure 1: Quantum Information Science is formed at the intersection of Computer Science and Quantum Physics

Whereas a typical introductory course to the subject develops the concepts from a computer science or information theoretic perspective (see Literature list), we want to take as starting point quantum theory as it is taught in a basic course on quantum physics at the Bachelor level. There, symmetry played a major role in analysing physical systems, so it is expected that quantum information processing can be analysed with similar methods.

We will illustrate this in the realm of quantum information processing with the fundamental problem of *entanglement distillation.* Entanglement stands for strong quantum correlations which we learn will help us with teleportation, superdense coding and cryptography. Entanglement distillation we call the process of extracting useful entanglement from not so useful entanglement. Since this is a process that involves many identical spins we will use the symmetry groups $SU(2)$ and the group of permutations responsible for particle exchange in order to analyse this process.

This will lead to us to the topic of quantum computation, where we will learn that computation can be performed by the permutation of particles. In order to obtain a universal quantum computer, however, fermions and bosons do not suffice and we need to take a look at the exchange of so-called anyons – leading us to the *topological model of quantum computation.*

1.2 Literature

Lecture notes will be handed out. The following is a list of literature on the broader topics which may be used as complementary reading.

- M. Nielsen and I. Chuang, Quantum Computation and Quantum Information, Cambridge University Press
- J. Preskill, Lecture Notes, Caltech, `http://www.theory.caltech.edu/people/preskill/ph229/references.html`
- R. Renner, Lecture Notes, ETH,
  `http://www.itp.phys.ethz.ch/education/lectures_fs10/QIT`

2 Formalism of Quantum Mechanics

In this section we will repeat the axiomatic approach to quantum theory, known from a basic course in quantum mechanics, with a view towards its application in quantum information theory.

2.1 Axioms

The following are the axioms of quantum mechanics.

- (System) To every physical system we associate a Hilbert space $\mathcal{H}$. (In this course, we will restrict attention to the finite dimensional case, i.e. $\mathcal{H} \cong \mathbb{C}^d$.) The Hilbert space associated to a joint system that is composed of two subsystems A and B with respective Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ is given by the tensor product $\mathcal{H}_A \otimes \mathcal{H}_B$.
- (State) The state of a system is described by a vector $|\psi\rangle \in \mathcal{H}$ that is normalised $\langle\psi|\psi\rangle = 1$. [1]
- (Time Evolution) The (discrete) time evolution of the state $|\psi\rangle$ of a system $\mathcal{H}$ corresponds to the application of a unitary matrix $U \in U(\mathcal{H})$, i.e. [2]

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

- (Measurement) Hermitian operators (observables) correspond to observable quantities. Given an observable R, consider its spectral decomposition $R = \sum_i r_i P_i$ where P_i is the projector onto the eigenspace of R with

[1] We make use of Dirac's bra-ket notation: For the column vector $\begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_d \end{pmatrix}$ we write $|\psi\rangle$ and for the corresponding row vector $(\bar{\psi}_1, \bar{\psi}_2, \ldots, \bar{\psi}_d)$ we write $\langle\psi|$. The natural inner product between $|\psi\rangle$ and $|\phi\rangle \in \mathcal{H}$ then takes the form $\langle\psi|\phi\rangle = \sum_i \bar{\psi}_i \phi_i$.

[2] For a time-independent Hamiltonian H and an evolution of time Δt, $U = e^{iH\Delta t}$.

eigenvalue r_i. (All r_i are distinct). The measurement results in outcome r_i with probability $p_i := \mathrm{tr} P_i |\psi\rangle\langle\psi|$. The post-measurement state given that result r_i was obtained is given by $\frac{1}{\sqrt{r_i}} P_i |\psi\rangle$.

2.2 Density Matrices

A careful look at the axioms reveals that the states $|\psi\rangle$ and $e^{i\phi}|\psi\rangle$ cannot be distinguished for any angle $\phi \in [0, 2\pi)$ by measurement (even if preceeded by a time evolution). The state of a system is therefore described by the equivalence class of states $\{e^{i\phi}|\psi\rangle, \phi \in [0, 2\pi)\}$, or equivalently the projector $|\psi\rangle\langle\psi|$, rather than the vector $|\psi\rangle$.

Imagine that we are given a source that ejects a particle in state $|\psi_j\rangle\langle\psi_j|$ with probability q_j but that we are not told the value j.

$$\boxed{\text{SOURCE}}\!-\ \{q_j, |\psi_j\rangle\langle\psi_j|\}$$

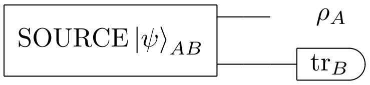
How should we describe the state of the system? Pragmatically we should find a description such that all measurements of this particle are described accurately. That is if we measure observable R, then from our description we should be able to compute that r_i occurs with probability $p_i = \sum_j q_j \mathrm{tr} P_i |\psi_j\rangle\langle\psi_j|$. Exchanging the sum and the trace we find $p_i = \mathrm{tr} P_i \left(\sum_j q_j |\psi_j\rangle\langle\psi_j|\right)$. If we therefore describe the state of the particle that the source emits by the operator $\rho := \sum_j q_j |\psi_j\rangle\langle\psi_j|$ we can compute all probabilities accurately.

$$\boxed{\text{SOURCE}}\!-\ \rho$$

Note that ρ is a positive semi-definite Hermitian operator with $\mathrm{tr}\rho = 1$ and that – by the spectral theorem – every such operator can be generated by some source. We therefore make the following definition: an operator ρ is called *density operator (or density matrix or state)* if it is a positive semi-definite Hermitian operator with trace equal to one. Note that a density operator ρ takes the form $|\psi\rangle\langle\psi|$ for some $|\psi\rangle$ if and only the rank of ρ equals to one. Such states are called *pure states*, states of higher rank are known as *mixed states.*

We have thus introduced density operators in order to account for our ignorance of the label j. There is another reason to introduce density operators: Say we are given two particles in state $|\psi\rangle\langle\psi|_{AB}$, where $|\psi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$ and would like to describe the state of the first system only. Again we want to make sure that the probabilities of a measurement with observable R on particle A are described accurately, $p_i = \mathrm{tr}(P_i \otimes \mathbf{1}_B)|\psi\rangle\langle\psi|_{AB}$. Executing the trace in two steps, first as a trace over system B and then over system A, we find $p_i = \mathrm{tr} P_i \rho_A$, where $\rho_A = \mathrm{tr}_B |\psi\rangle\langle\psi|_{AB} := \sum_{kl} |k\rangle\langle l|_A \mathrm{tr}[|l\rangle\langle k|_A \otimes \mathbf{1}_B |\psi\rangle\langle\psi|_{AB}]$, where $|k\rangle_A$ is an orthonormal basis for $\mathcal{H}_A$. Since $|\psi\rangle\langle\psi|_{AB}$ is positiv semidefinite with $\mathrm{tr}|\psi\rangle\langle\psi|_{AB} = 1$ it follows that also ρ_A is positiv semidefinite with $\mathrm{tr}\rho_A = 1$ and hence a density matrix. The following simple example shows that ρ_A in general does not have rank one and hence is not of the form $\rho_A = |\phi\rangle\langle\phi|_A$ for $|\phi\rangle_A \in \mathcal{H}_A$. As an example consider $\mathcal{H}_A \cong \mathcal{H}_B \cong \mathbb{C}^2$ with $|\psi\rangle_{AB} :=$

$\frac{1}{\sqrt{2}}(|0\rangle_A |1\rangle_B - |1\rangle_B |0\rangle_A)$, the singlet state (we typically omit the tensor product, i.e. $|0\rangle |1\rangle = |0\rangle \otimes |1\rangle$). A simple calculation shows that $\rho_A = \frac{\mathbf{1}_A}{2}$, hence rank $\rho_A = 2$.



Given a density matrix ρ_{AB} we can of course also compute the partial trace over system B.

3 The Qubit

3.1 The Bloch sphere

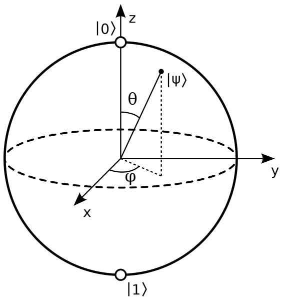
The simplest quantum system has a two-dimensional state space $\mathcal{H} \cong \mathbb{C}^2$, the qubit. We know it already quite well as the spin-$\frac{1}{2}$ system, and so we start by introducing the Pauli matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Together with the identity the Pauli matrices form a basis for the two-by-two Hermitian matrices. Since density matrices are Hermitian and have non-negative eigenvalues that sum to one, we can express any density operator ρ as

$$\rho = \frac{1}{2}(\mathbf{1} + \vec{v}.\vec{\sigma}),$$

for the *Bloch vector* $\vec{v} \in \mathbb{R}^3$ with $v_x^2 + v_y^2 + v_z^2 \leq 1$, $\sigma = (\sigma_x, \sigma_y, \sigma_z)$ and $\vec{v}.\vec{\sigma} = v_x\sigma_x + v_y\sigma_y + v_z\sigma_z$. A density matrix can thus be represented as a vector in the *Bloch ball*, the three dimensional unit ball. When $v_x^2 + v_y^2 + v_z^2 = 1$, the density matrix has rank one and the density matrix (or vector) lies in the *Bloch sphere*. The graphics shows the state $|\psi\rangle$ parametrised as $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle, \theta \in [0, \pi), \phi \in [0, 2\pi)$. The Bloch vector is $\vec{v} = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$.



3.2 $SU(2)$ versus $SO(3)$

The group $SU(2)$ is the group of two-by-two unitary matrices with determinant one, i.e. it consists of the elements

$$\begin{pmatrix} \alpha & -\bar{\beta} \\ \beta & \bar{\alpha} \end{pmatrix}$$

where $\alpha, \beta \in \mathbb{C}$ with $|\alpha|^2 + |\beta|^2 = 1$. From this parametrisation, it is obvious that as a manifold, $SU(2) \cong S^3$, the three-sphere, since for $\alpha = x + iy$ and $\beta = z + iw$, with $x, y, z, w \in \mathbb{R}$: $x^2 + y^2 + z^2 + w^2 = 1$.

$SU(2)$ is a Lie group and as such one can compute its Lie algebra (the tangent space at the point $\mathbf{1} \in SU(2)$. In order to do so we note that the one parameter subgroups are of the form e^{itA} with A traceless and Hermitian. The derivative at $\mathbf{1}$ then shows that the Lie algebra $su(2)$ of $SU(2)$ consists of traceless Hermitian matrices, i.e. is spanned by Pauli matices. The latter satisfy the commutation relations

$$[\sigma_x, \sigma_y] = 2i\sigma_z. \tag{1}$$

We can then look at the one-parameter subgroups of $SU(2)$ as given by rotations around a unit vector $\vec{e}$ by an angle $\alpha \in [0, 4\pi)$: $U(\vec{e}, \alpha) = e^{-i\frac{\alpha}{2}\vec{e}.\vec{\sigma}} = \cos\frac{\alpha}{2}\mathbf{1} - i\sin\frac{\alpha}{2}\vec{e}.\vec{\sigma}$. This unitary matrix when applied to a density matrix (by conjugation) results in a rotation $R(\vec{e}, \alpha)$ of the Bloch vector around the axis $\vec{e}$ with an angle α in the Bloch sphere according to the formula

$$U(\vec{e}, \alpha)\,(\vec{v}.\vec{\sigma})\,U(\vec{e}, \alpha)^\dagger = (R(\vec{e}, \alpha).\vec{v}).\vec{\sigma}.$$

The map

$$SU(2) \ni U(\vec{e}, \alpha) \mapsto R(\vec{e}, \alpha) \in SO(3)$$

is a homomorphism with kernel $\{\mathbf{1}, -\mathbf{1}\}$. $SU(2)$ is the double covering group of $SO(3)$, the Lie algebras are isomorphic. The fact that $SU(2)$ is the (simply connected) double cover of the (not simply connected) $SO(3)$ can be illustrated nicely with Dirac's spinner-spanner (see also [2]). [3]

3.3 Measurement

We can measure a qubit along a certain direction in space, given by a unit vector $\vec{e}$. With this we mean that we measure the observable $\vec{e}.\vec{\sigma}$. Note that it is sufficient to consider traceless observables and unit vectors, as the addition of identity and the stretching of the vector does not effect the statistics, but

[3] An element in $SO(3)$ is given by an angle and an axis, for instance represented as a globe (north south axis and point, e.g. Rome.) A path in SO(3) can thus be represented as the movement of the globe with the arm (the center does not play a role). A path homotopic to the trivial path (no movement) is thus a wiggle of the arm. Interestingly we cannot obtain a single revolution of the globe around a fixed axis (a path that starts at the identity and ends as the identity) as a wiggle of the arm, but we can with a double revolution. Hence $SO(3)$ is not simply connected but has a simply connected double cover group.

only the labeling of the outcomes. The projectors onto the two eigenvectors are given by $P_\pm := \frac{1}{2}(\mathbf{1} \pm \vec{e}.\vec{\sigma})$. The probabilities are

$$\mathrm{tr} P_\pm \rho = \frac{1}{2}(1 \pm \vec{e}.\vec{v})$$

4 The Entangled Bit or Ebit

4.1 Superdense Coding

Here we will explain how Alice can send *two classical bits* of information to Bob by sending only *one qubit* under the assumption that they share an ebit, that is the state

$$\frac{1}{\sqrt{2}} |00 + 11\rangle_{AB} .$$

This is known as *superdense coding.*

Before we explain how this works, let us introduce the four *Bell states* $|\psi_{\alpha\beta}\rangle$

$$|\psi_{00}\rangle = \frac{1}{\sqrt{2}} |00 + 11\rangle$$

$$|\psi_{01}\rangle = \frac{1}{\sqrt{2}} |00 - 11\rangle$$

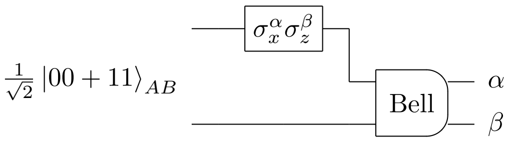
$$|\psi_{10}\rangle = \frac{1}{\sqrt{2}} |01 + 10\rangle$$

$$|\psi_{11}\rangle = \frac{1}{\sqrt{2}} |01 - 10\rangle$$

Note that $\sigma_x^\alpha \sigma_z^\beta \otimes \mathbf{1}_B |\psi_{00}\rangle = |\psi_{\alpha\beta}\rangle$. The *Bell measurement* is the joint measurement of the commuting operators $\sigma_z \otimes \sigma_z$ and $\sigma_x \otimes \sigma_x$. Since the joint eigenstates of these two operators are the Bell states, the measurement projectors are the projectors onto these states.

The superdense coding protocol then works as follows.

- Alice wants to send two bits $\alpha, \beta \in \{0, 1\}$ to Bob.
- Alice and Bob have systems A and B, respectively in a state $\frac{1}{\sqrt{2}} |00 + 11\rangle_{AB}$
- Alice applies the unitary (Pauli) matrix $\sigma_x^\alpha \sigma_z^\beta$ to system A.
- Alice sends qubit A to Bob
- Bob performs the Bell measurement on systems AB and obtains outcome $\alpha, \beta \in \{0, 1\}$.


$\sigma_x^\alpha \sigma_z^\beta$
$\frac{1}{\sqrt{2}} |00 + 11\rangle_{AB}$
Bell
α
β

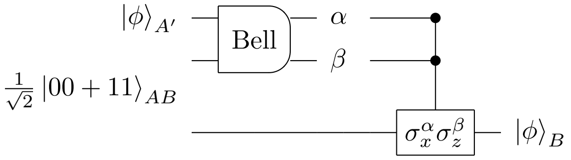
The protocol clearly transmits the two bits correctly.

4.2 Teleportation

"An unknown quantum state $|\phi\rangle$ can be disassembled into, then later reconstructed from, purely classical information and purely nonclassical EPR correlations. To do so the sender, Alice, and the receiver, Bob, must prearrange the sharing of an EPR-correlated pair of particles. Alice makes a joint measurement on her EPR particle and the unknown quantum system, and sends Bob the classical result of this measurement. Knowing this, Bob can convert the state of his EPR particle into an exact replica of the unknown state $|\phi\rangle$ which Alice destroyed." [3]

In other words we have the following setup

- Alice has a system A' in state $|\phi\rangle_{A'}$ that she wants to transfer to Bob.
- Alice and Bob have systems A and B, respectively in a state $\frac{1}{\sqrt{2}}\,|00+11\rangle_{AB}$
- Alice performs the Bell measurement (see below) on systems AA' and obtains outcome $\alpha, \beta \in \{0,1\}$.
- Alice tells Bob what outcome she has obtained and he applies the unitary (Pauli) matrix $\sigma_x^\alpha \sigma_z^\beta$ to system B.
- Bob holds state $|\phi\rangle_B$.


$|\phi\rangle_{A'}$
Bell
α
β
$\frac{1}{\sqrt{2}}\,|00+11\rangle_{AB}$
$\sigma_x^\alpha \sigma_z^\beta$
$|\phi\rangle_B$

Note that $\mathrm{tr}_B|\psi_{\alpha\beta}\rangle\langle\psi_{\alpha\beta}| = \frac{1}{4}\mathbf{1}_A$ for all α, β. Each outcome is equally probable, since

$$\begin{aligned}\mathrm{tr}(|\psi_{\alpha\beta}\rangle\langle\psi_{\alpha\beta}|_{A'A} \otimes \mathbf{1}_B)(|\phi\rangle\langle\phi|_{A'} \otimes |\psi_{00}\rangle\langle\psi_{00}|_{AB}) \\ = \mathrm{tr}|\psi_{\alpha\beta}\rangle\langle\psi_{\alpha\beta}|_{A'A}(|\phi\rangle\langle\phi|_{A'} \otimes \frac{1}{2}\mathbf{1}_A) \\ = \frac{1}{2}\mathrm{tr}\frac{1}{2}\mathbf{1}_{A'}|\phi\rangle\langle\phi|_{A'} = \frac{1}{4}\end{aligned}$$

Let us now write $|\phi\rangle = c_0\,|0\rangle + c_1\,|1\rangle$. The state on Bob's side, given that Alice has measured $\alpha\beta$ is given by

$$\begin{aligned}
\langle\psi_{\alpha\beta}|\,(c_0\,|0\rangle\,+c_1\,|1\rangle)\,|\psi_{00}\rangle \\
&= c_0\,\langle\psi_{\alpha\beta}|\,|0\rangle\,|\psi_{00}\rangle + c_1\,\langle\psi_{\alpha\beta}|\,|1\rangle\,|\psi_{00}\rangle \\
&= \frac{1}{\sqrt{2}}(c_0\langle\psi_{\alpha\beta}|000+011\rangle + c_1\langle\psi_{\alpha\beta}|100+111\rangle) \\
&= \frac{1}{2\sqrt{2}}(c_0\langle\psi_{\alpha\beta}|(00+11)0+(00-11)0+(01+10)1+(01-10)1\rangle \\
&\quad + c_1\langle\psi_{\alpha\beta}|(10+01)0+(10-01)0+(11+00)1+(11-00)1\rangle) \\
&= \frac{1}{2}\big(c_0\,\langle\psi_{\alpha\beta}|\,(|\psi_{00}\rangle\,|0\rangle+|\psi_{01}\rangle\,|0\rangle+|\psi_{10}\rangle\,|1\rangle+|\psi_{11}\rangle\,|1\rangle) \\
&\quad + c_1\,\langle\psi_{\alpha\beta}|\,(|\psi_{10}\rangle\,|0\rangle-|\psi_{11}\rangle\,|0\rangle+|\psi_{00}\rangle\,|1\rangle-|\psi_{01}\rangle\,|1\rangle)\,\big) \\
&= \frac{1}{2}\big(c_0\,|\alpha\rangle+(-1)^{\beta}c_1\,|\bar{\alpha}\rangle\,\big) \\
&= \frac{1}{2}\sigma_x^{\alpha}\sigma_z^{\beta}\,|\phi\rangle
\end{aligned}$$

Bob now applies $\sigma_x^{\alpha}\sigma_z^{\beta}$ after which the state is back in the state $|\phi\rangle$ since the Pauli matrices square to the identity. Note that the transmission of the classical bits is essential, otherwise the state is $\mathbf{1}/4$. Since every mixed state of one qubit is a mixture of pure states, the same holds for true for mixed states by linearity.

This illustrates that the ebit is a great resource in quantum information theory. Other uses of ebits are in quantum key distribution.

4.3 Distilling Ebits

Unfortunately, it is not so easy to distribute ebits between Alice and Bob. In practice, namely, imperfect transmission channels (e.g. glas fibres) introduce noise. The following question arises:

"Given a state ρ_{AB}, can we convert it into an ebit by application of local transformation and classical communication?"

We have seen before that for instance when we are given $\frac{1}{\sqrt{2}}\,|01+10\rangle$ we can convert it into $\frac{1}{\sqrt{2}}\,|00+11\rangle$ by application of σ_z on Alice's or Bob's side only. In general this is not possible, so we have to refine our question.

"Given n copies of a state ρ_{AB}, how many ebits can we extract by application of local transformation and classical communication per copy when $n\to\infty$?"

Since this is a question about states that live on $(\mathcal{H}_A\otimes\mathcal{H}_B)^{\otimes n}$ where n is large, we will have to develop tools in order to deal with them. The tools that we develop in this course will given an answer to this question for pure states $\rho_{AB}=|\psi\rangle\langle\psi|_{AB}$.

5 Spin

In this section, we will repeat, introduce and develop the representation-theoretic tools necessary in order to solve the entanglement distillation problem introduced above. For simplicity, we will only do this for qubits, i.e. the group $SU(2)$. All this can also be done in higher dimensions, i.e. for $SU(d)$. Since this will complicate the proofs significantly, we will not do this in this lecture. The interested student is referred to the PhD theses of Aram Harrow (arXiv:quant-ph/0512255) and myself (arXiv:quant-ph/0604183).

5.1 Group Representations

Let us quickly recall that a *representation* of a group G on a (complex) vector space V is a map

$$T : G \to GL(V)$$

that preserves the group operation, i.e. for all $g, h \in G$

$$T(g)T(h) = T(gh).$$

$\dim V$ is called the dimension of the representation. A representation is *irreducible* if the only subspaces of V that are invariant under G are the empty subspace and V itself.

Theorem 1. *Let T be a representation of a finite group G. Then T is isomorphic to a direct sum of irreducible representations of G, i.e. $T \cong \bigoplus_i T_i$ for irreducible representations T_i of G.*

Proof. Let W be the vector space on which G acts. Let (w_1, w_2) be a scalar product on W, then

$$\{w_1, w_2\} = \frac{1}{|G|} \sum_{g \in G} (gw_1, gw_2)$$

is a G-invariant scalar product on W. If V is an invariant subspace of W, then $V^\perp$, the orthogonal complement of V in W, is also an invariant subspace: for $v \in V$ and $v^\perp \in V^\perp$, $\{gv^\perp, v\} = \{v^\perp, g^{-1}v\} = 0$, since $g^{-1}v \in V$ and V is G-invariant. In this way one can keep on breaking up the space of W into invariant subspaces. This procedure will terminate, because W is finite-dimensional. □

In fact this theorem holds for any compact group, since their exists a unique invariant probability measure on the group (the Haar measure) so that we can perform the averaging trick. In the exercises the Haar meausure for $SU(2)$ will be constructed explicitely.

Two representations T_1 and T_2 are equivalent if they have the same dimension and if there is an element $K \in GL(V)$ such that for all $g \in G$: $KT_1(g)K^{-1} = T_2(g)$. Whenever the averaging trick works, then any representation T_1 is equivalent to a unitary representation,

$$T_2 : G \to U(V).$$

In order to see this write explicitly $\{v,w\} = \langle v| A |v\rangle$ for some positive-definite matrix A. Note that $A = K^\dagger K$ for an invertible K. Then (defining $|w'\rangle = K|w\rangle$ and $|v'\rangle = K|v\rangle$)

$$\begin{aligned}\langle v|w\rangle = \langle v'| K^\dagger K |w'\rangle &= \{v',w'\} = \{T_1(g)v', T_1(g)w'\} \\ &= \langle v'| T_1(g)^\dagger K^\dagger K T_1(g) |w'\rangle \\ &= \langle v| (K^{-1})^\dagger T_1(g)^\dagger K^\dagger K T_1(g) K^{-1} |w\rangle \\ &= \langle v| T_2(g)^\dagger T_2(g) |w\rangle .\end{aligned}$$

We denote with $\hat{G}$ the set of equivalence classes of representations of G. According to Theorem1, a representation T of G can be decomposed into a direct sum of irreducible representations:

$$T \cong \bigoplus_{\alpha\in\hat{G}} T_\alpha \otimes \mathbf{1}_{m_\alpha}.$$

Each irreducible representation T_α occurs with multiplicity $m_\alpha \in \mathbb{N}_0$. $\mathbf{1}_d$ denotes the identity matrix on $\mathbb{C}^d$.

Given two representations $T_1 : G \to GL(V_1)$ and $T_2 : G \to GL(V_2)$ we define the tensor product representation

$$T_1 \otimes T_2 : G \to GL(V_1 \otimes V_2)$$

by

$$(T_1 \otimes T_2)(g) := T_1(g) \otimes T_2(g).$$

Tensor product representations are reducible in general.
Probably the most frequently used result in representation theory is the famous lemma by Isaac Schur.

Lemma 2 (Schur's lemma)**.** *Let T_1 and T_2 be irreducible representations of G acting on V_1 and V_2, respectively. If the homomorphism $\phi : V_1 \to V_2$ commutes with the action of G, then*

- *either ϕ is an isomorphism, or $\phi = 0$.*
- *if $V_1 = V_2$, then $\phi = \lambda\mathbf{1}$ for some $\lambda \in \mathbb{C}$.*

Proof. $\ker\phi$ is an invariant subspace of V_1 since

$$\begin{aligned}|v\rangle \in \ker\phi &\Rightarrow \phi|v\rangle = 0 \\ &\Rightarrow T_2(g)\phi|v\rangle = 0 \\ &\Rightarrow \phi T_1(g)|v\rangle = 0 \\ &\Rightarrow T_1(g)|v\rangle \in \ker\phi.\end{aligned}$$

$\mathrm{im}\phi$ is an invariant subspace of V_2 since

$$\begin{aligned}|v\rangle \in \mathrm{im}\phi &\Rightarrow \exists\, |w\rangle \in V_1; |v\rangle = \phi|w\rangle \\ &\Rightarrow T_2(g)|v\rangle = T_2(g)\phi|w\rangle = \phi T_1(g)|w\rangle \\ &\Rightarrow T_2(g)|v\rangle \in \mathrm{im}\phi.\end{aligned}$$

Since the action of G on V_1 and V_2 is irreducible, $\ker \phi, \mathrm{im}\phi \in \{\emptyset, V_1\}$. Hence ϕ is either an isomorphism or ϕ vanishes. This proves the first part of the lemma. Since $\mathbb{C}$ is algebraically closed, the characteristic polynomial $\det(\phi - \lambda \mathbf{1})$ must have a root $\lambda \in \mathbb{C}$ and hence $\ker(\phi - \lambda\mathbf{1}) \neq \emptyset$. $\phi - \lambda\mathbf{1}$ is therefore not an isomorphism, which implies (since $\phi - \lambda\mathbf{1}$ also commutes with the action of G) by the first part of the lemma that $\phi - \lambda\mathbf{1} = 0$. □

Schur's lemma implies that the decomposition in Theorem 1 is unique up to isomorphism. The classification of representations of a finite group G is therefore reduced to the classification of all irreducible representations.

Most theorems in this section carry over almost unchanged to compact groups. Most importantly this is true for Theorem 1 and Schur's lemma, Lemma 2.

In the following we will remind ourselves of the representation theory for $SU(2)$.

5.2 Representations of $SU(2)$

Since $SU(2)$ is compact the above discussion applies and we can assume that all representations are unitary, in fact with determinant one. Since $SU(2)$ is furthermore simply connected, the irreducible representations stand in one-to-one relation with irreducible representations of its Lie algebra $su(2)$[4] A representation of $su(2)$ is a homomorphism

$$t : su(2) \to \mathrm{End}(V)$$

that preserves the Lie bracket

$$[t(A), t(B)] = t([A, B]).$$

In particular, we find with (1),

$$[t(\sigma_x), t(\sigma_y)] = 2it(\sigma_z).$$

This representation extends linearly to

$$su(2)_{\mathbb{C}} := su(2) \oplus isu(2) \cong sl(2) = \{\text{2x2 complex traceless matrices}\}.$$

It is often convenient to consider $su(2)_{\mathbb{C}}$ instead of $su(2)$ since it can be given a somewhat nice bases in terms of the raising and lowering operators and the Pauli-σ_z matrix:

$$\sigma_+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \qquad \sigma_- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

[4] A Lie algebra $\mathfrak{g}$ is a (real or complex) vector space together with a bilinear map (Lie bracket) $[,] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ that satisfies

- (linearity $[\alpha a + \beta b, c] = \alpha[a, c] + \beta[b, c]$
- (skew-symmetry]) $[a, b] = -[b, a]$
- (Jacobi) $[a, [b, c]] + [b, [c, a]] + [c, [a, b]] = 0$

for all $a, b, c \in \mathfrak{k}$ and real or complex α and β.

We now quickly recall the irreducible representations of $SU(2)$. For each $k \in \mathbb{N}_0$, there is a $k+1$ dimensional irreducible representation, in the physics literature known as spin-$\frac{k}{2}$ representation. We define this representation, which we denote by $\mathcal{V}_k$ as an $SU(2)$ and as v_k as a representation of $su(2)$ by its action on the $k+1$ orthonormal basis states $|k,l\rangle$, $0 \leq l \leq k$. The space on which these representations act is denoted by V_k.

$$v_k(\sigma_-)\,|k,l\rangle = \sqrt{l(k-l+1)}\,|k,l-1\rangle$$

$$v_k(\sigma_+)\,|k,l\rangle = \sqrt{(k-l)(l+1)}\,|k,l+1\rangle$$

$$v_k(\sigma_z)\,|k,l\rangle = (2l-k)\,|k,l\rangle$$

$2l-k$ is called the *weight* of the *weight vector* $|k,l\rangle$. From which follows that the total angular momentum (or Casimir) operator is a scalar:[5]

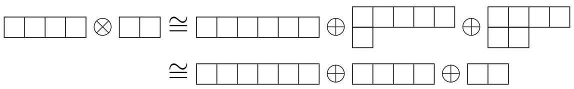
$$\sum_{i\in\{x,y,z\}} v_k(\sigma_i)v_k(\sigma_i) = k(k+2)\mathbf{1}. \tag{2}$$

5.3 Clebsch-Gordan Decomposition

The decomposition of two irreducible representations of $SU(2)$ is known as the Clebsch-Gordan decomposition. It reads

$$V_{k_1} \otimes V_{k_2} \cong \bigoplus_{k=|k_1-k_2|:k \bmod 2=k_1+k_2 \bmod 2}^{k_1+k_2} V_k.$$

In other words the multiplicities m_k are one exactly when $|k_1-k_2| \leq k \leq k_1+k_2$ with k even if k_1+k_2 even and zero otherwise. We illustrate this decomposition in terms of so-called Young frames.



The intermediate step corresponds to $U(2)$ representations, which ”remember“ the number of boxes.

Writing down the change of basis explicitly

$$|k,l\rangle = \sum_{l_1,l_2} |k_1,l_1\rangle\,|k_2,l_2\rangle\,(\langle k_1,l_1|\,\langle k_2,l_2|\,|k,l\rangle)\,. \tag{3}$$

[5] In general, one can replace the sum over Pauli operators by a sum over the elements of a basis of the Lie algebra and the left argument by the corresponding elements from the dual basis. This gives us the alternative formula

$$2v_k(\sigma_-)v_k(\sigma_+) + 2v_k(\sigma_+)v_k(\sigma_-) + v_k(\sigma_z)v_k(\sigma_z) = k(k+2)\mathbf{1}$$

The coefficients $\langle k_1, l_1| \langle k_2, l_2| |k, l\rangle$ are known as *Clebsch-Gordan coefficients* and form a unitary matrix. One can find them in many books and we will only recall the important special case where $k_2 = 1$. Our notation reduces in this case to $|1, 1\rangle = |1\rangle$ and $|1, 0\rangle = |0\rangle$.

Before we do so, let us not that taking the derivative of the action of $g \in SU(2)$ on $V_1 \otimes V_2$ we obtain the following action for $su(2)$ on $V_1 \otimes V_2$:

$$su(2) \ni a \mapsto (v_{k_1} \otimes v_{k_2})(a) = v_{k_1}(a) \otimes \mathbf{1} + \mathbf{1} \otimes v_{k_2}(a).$$

This implies that the weight of the vectors is additive, i.e. that we can restrict the sum in eq.(3) to $2l_1 - k_1 + 2l_2 - k_2 = 2l - k$. In particular we find

$$|k+1, k+1\rangle = |k, k\rangle |1, 1\rangle .$$

The vector $|k+1, k\rangle$ is now obtained applying the lowering operator to this equation. Hence

$$v_{k+1}(\sigma_-) |k+1, k+1\rangle = (v_k(\sigma_-) |k, k\rangle) |1, 1\rangle + |k, k\rangle v_1(\sigma_-) |1, 1\rangle$$

or

$$|k+1, k\rangle = \sqrt{\frac{k}{k+1}} |k, k-1\rangle |1, 1\rangle + \sqrt{\frac{1}{k+1}} |k, k\rangle |1, 0\rangle$$

and by induction we find

$$|k+1, l\rangle = \sqrt{\frac{l}{k+1}} |k, l-1\rangle |1, 1\rangle + \sqrt{\frac{k+1-l}{k+1}} |k, l\rangle |1, 0\rangle$$

The states $|k-1, l-1\rangle$ then follow (up to a phase factor) since they have to be orthogonal to $|k+1, l\rangle$:

$$|k-1, l-1\rangle = -\sqrt{\frac{k+1-l}{k+1}} |k, l-1\rangle |1, 1\rangle + \sqrt{\frac{l}{k+1}} |k, l\rangle |1, 0\rangle .$$

Inverting this transformation and remembering with help of a path label $p \in \{1, -1\}$ whether we have increased or decreased the total spin we find

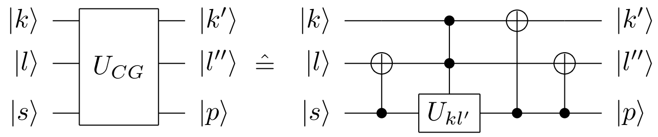
$$|k, l\rangle |1, 0\rangle = \sqrt{\frac{k+1-l}{k+1}} |k+1, l, +\rangle + \sqrt{\frac{l}{k+1}} |k-1, l-1, -\rangle$$

$$|k, l-1\rangle |1, 1\rangle = \sqrt{\frac{l}{k+1}} |k+1, l, +\rangle - \sqrt{\frac{k+1-l}{k+1}} |k-1, l-1, -\rangle .$$

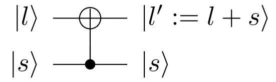
Defining the unitary matrix

$$U_{kl} := \begin{pmatrix} \sqrt{\frac{l}{k+1}} & \sqrt{\frac{k+1-l}{k+1}} \\ -\sqrt{\frac{k+1-l}{k+1}} & \sqrt{\frac{l}{k+1}} \end{pmatrix}$$

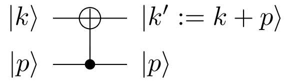
which corresponds to a rotation around the y-axis with angle $\theta_{kl} := \arccos\sqrt{\frac{l}{k+1}}$, we find the following circuit transforming the spin information[6] s into path information[7] p [1].



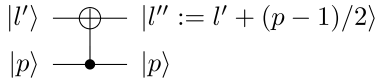
where



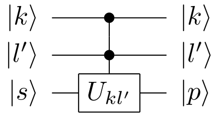
and



and



and



The circuits are to be understood in the following way:

- the vector $|k,l\rangle = |k\rangle\,|l\rangle$ for $0 \leq l \leq k$
- the circuit is only defined on valid inputs, i.e. $0 \leq l \leq k$
- the registers holding k and l are assumed to be big enough, so that all computations can be performed (i.e. $\mathbb{C}^n$ for $k, k', l, l' \leq n$.)
- the Clebsch-Gordan circuit defines an isometry from the space of valid inputs to the output space $\mathbb{C}^n \otimes \mathbb{C}^n \otimes \mathbb{C}^2$, holding k', l' and p.

5.4 Schur Transform

In this section, we want to iteratively decompose $V_1^{\otimes n}$ into irreducible representations. We will do this by applying consecutively the Clesch-Gordan decom-

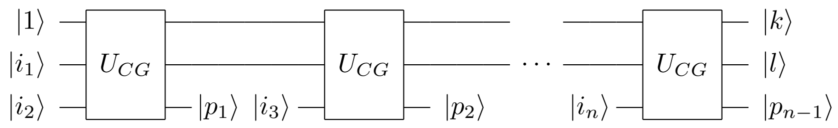
[6] $|1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

[7] $|+\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|-\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

position.

$$\begin{aligned}V_1^{\otimes n} &\cong (V_0 \oplus V_2) \otimes V_1^{\otimes n-2} \\ &\cong (V_0 \otimes V_1 \oplus V_2 \otimes V_1) \otimes V_1^{\otimes n-3} \\ &\cong (V_1 \oplus (V_1 \oplus V_3)) \otimes V_1^{\otimes n-3} \\ &\cong \big(V_1 \otimes \mathbb{C}^2 \oplus V_3\big) \otimes V_1^{\otimes n-3} \\ &\vdots \\ &\cong \bigoplus_k V_k \otimes \mathbb{C}^{m_k^n}\end{aligned}$$

The corresponding circuit is a concatenation of the circuits above [1].



Let $|p\rangle = |p_1\rangle \dots |p_{n-1}\rangle$ and note that the final states $|k\rangle |l\rangle |p\rangle$ take values $0 \leq k \leq n$, $0 \leq l \leq n$ and $p_i \in \{+, -\}$. Of course the number of possibilities for l is constraint to $0 \leq l \leq k$. For p there are m_k^n possibilities. Since all inputs are valid, the Schur transform is an isometry from $(\mathbb{C}^2)^{\otimes n}$ to $\mathbb{C}^n \otimes \mathbb{C}^n \otimes (\mathbb{C}^2)^{\otimes n-1}$.

5.5 The Number of Paths

Let us now compute a recursion formula for m_k^n and start by noting that n even implies $m_k^n = 0$ for k odd and vice versa. It is also clear that $m_k^n = 0$ for $k > n$. Assume

$$V_1^{\otimes n} \cong \bigoplus_{k=0}^{n} V_k \otimes \mathbb{C}^{m_k^n}$$

then

$$\begin{aligned}V_1^{\otimes n+1} &\cong \bigoplus_k V_k \otimes V_1 \otimes \mathbb{C}^{m_k^n} \\ &\cong \bigoplus_k (V_{k+1} \oplus V_{k-1}) \otimes \mathbb{C}^{m_k^n} \\ &\cong \bigoplus_k V_k \otimes \mathbb{C}^{m_{k-1}^n + m_{k+1}^n}\end{aligned}$$

where we defined $m_{-1}^n = 0$ for all n. We find that for $n \mod 2 = k + 1 \mod 2$:

$$m_k^{n+1} = m_{k-1}^n + m_{k+1}^n.$$

The multiplicities then follow from this formula and the base case $m^1_k = \delta_{k,1}$. We see that for $0 \leq k < n$ with $n \mod 2 = k \mod 2$

$$m^n_k := \binom{n}{\frac{n-k}{2}} - \binom{n}{\frac{n-k-2}{2}},$$

and $m^n_n = 1$ (and zero otherwise) satisfies the recursion relation since since

$$\begin{aligned} m^n_{k-1} + m^n_{k+1} &= \binom{n}{\frac{n-k+1}{2}} - \binom{n}{\frac{n-k-1}{2}} + \binom{n}{\frac{n-k-1}{2}} - \binom{n}{\frac{n-k-3}{2}} \\ &= \left[\binom{n}{\frac{n-k+1}{2}} + \binom{n}{\frac{n-k-1}{2}}\right] - \left[\binom{n}{\frac{n-k-1}{2}} + \binom{n}{\frac{n-k-3}{2}}\right] \\ &\overset{Pascal}{=} \binom{n+1}{\frac{n+1-k}{2}} - \binom{n+1}{\frac{n+1-k-2}{2}} \\ &= m^{n+1}_k \end{aligned}$$

where we used Pascal's rule. Let us briefly discuss the asymptotic behaviour of m^n_k when n and k are large. Note that

$$m^n_k = \binom{n}{\frac{n-k}{2}} \frac{2k+2}{n+k+2}. \tag{4}$$

Note that for all $\alpha \in [0,1]$.

$$1 = (\alpha + (1-\alpha))^n = \sum_{j=1}^{n} \alpha^j (1-\alpha)^{n-j} \binom{n}{j}$$

The function $f(j) := \alpha^j (1-\alpha)^{n-j} \binom{n}{j}$ is thus a probability distribution and peaks when $j \approx n\alpha$. This is illustrated in Figure 2.

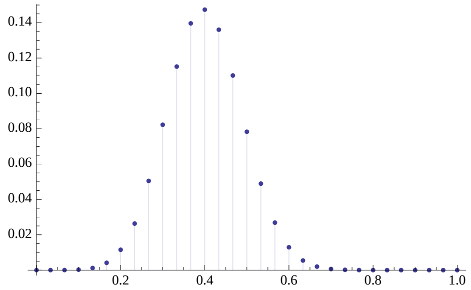


Figure 2: Plot of $f(j) := \alpha^j (1-\alpha)^{n-j} \binom{n}{j}$ for $\alpha = 0.4$ and $n = 30$.

Let now j_0 be an integer. Since there are no more than $n+1$ values for j, we find for $\alpha_0 := \frac{j}{n}$

$$\alpha^{j_0} (1-\alpha)^{n-j_0} \binom{n}{j_0} \geq \frac{1}{n+1}.$$

or

$$\log \binom{n}{j_0} \geq nh\left(\frac{j_0}{n}\right) - \log(n+1).$$

where $h(x) = -x\log x - (1-x)\log(1-x)$ is the binary entropy function (all logarithms are to base two).

Conversely (this time for all j, in particular j_0)

$$\alpha^{j_0}(1-\alpha)^{n-j_0}\binom{n}{j_0} \leq 1$$

or

$$\log \binom{n}{j_0} \leq nh\left(\frac{j_0}{n}\right)$$

In summary

$$nh\left(\frac{j_0}{n}\right) - \log(n+1) \leq \log\binom{n}{j_0} \leq nh\left(\frac{j_0}{n}\right).$$

Inserting $j_0 = (n-k)/2$, which is always an integer since n and k have the same parity:

$$nh\left(\frac{1}{2}(1-\frac{k}{n})\right) - 2\log(n+1) \leq \log m_k^n \leq nh\left(\frac{1}{2}(1-\frac{k}{n})\right)$$

That is, m_k^n is growing exponentially in n (if k is linear in n): $m_k^n \approx 2^{nh(\frac{1}{2}(1-\frac{k}{n}))+O(\log n)}$.

5.6 Schur-Weyl Duality

We have seen in the previous section how $SU(2)$ acts on density operators by rotating the Bloch vector. Let us go one step back and note that $SU(2)$ acts on vectors in $\mathbb{C}^2$ simply by left multiplication, this means that

$$\mathcal{V}_1 : SU(2) \to SU(2)$$

with $\mathcal{V}_1(g) = g$. It is clear that all the group operations are preserved and we therefore have a representation of $SU(2)$. It is called the *defining* representation of $SU(2)$. It is easy to verify that it is irreducible.

Recall the n-fold tensor product

$$\mathcal{V}_1^{\otimes n} : SU(2) \to SU(2^n)$$

$$\mathcal{V}_1^{\otimes n}(g) = g^{\otimes n}.$$

A basis for the space $\mathbb{C}^{2^{\otimes n}}$ on which this representation acts is given by

$$|i_1 i_2 \dots i_n\rangle := |i_1\rangle\,|i_2\rangle \dots |i_n\rangle = |i_1\rangle \otimes |i_2\rangle \dots |i_n\rangle$$

where $\{|1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}\}$ is a basis for $\mathbb{C}^2$.

There is a second, very natural action on the tensor space. Namely that of the symmetric group S_n permuting the tensor factors

$$\pi \left| i_1 \dots i_n \right\rangle = \left| i_{\pi^{-1}(1)} \dots i_{\pi^{-1}(n)} \right\rangle .$$

This is a representation of S_k since

$$\begin{aligned} \pi' \pi \left| i_1 \dots i_n \right\rangle &= \pi' \left| i_{\pi^{-1}(1)} \dots i_{\pi^{-1}(n)} \right\rangle \\ &=: \pi' \left| j_1 \dots j_n \right\rangle \\ &= \left| j_{\pi'^{-1}(1)} \dots j_{\pi'^{-1}(n)} \right\rangle \\ &=: \left| j_{\ell_1} \dots j_{\ell_n} \right\rangle \\ &= \left| i_{\pi^{-1}(\ell_1)} \dots i_{\pi^{-1}(\ell_n)} \right\rangle \\ &= \left| i_{\pi^{-1}(\pi'^{-1}(1))} \dots i_{\pi^{-1}(\pi'^{-1}(n))} \right\rangle \\ &= \left| i_{(\pi'\pi)^{-1}(1)} \dots i_{(\pi'\pi)^{-1}(n)} \right\rangle \end{aligned}$$

It is easy to see that this action commutes with the action of $SU(2)$ on this space. Hence, it acts only on the multiplicity space, the space of paths. In fact it acts irreducibly on each component. This is a consequence of the following lemma

Lemma 3.

$$\operatorname{span}_{\mathbb{C}}(A^{\otimes n}) = \{X : [X, \pi] = 0 \ \forall \ \pi \in S_n\}.$$

Proof. Clearly the LHS is contained in the RHS. We now turn to the proof that the RHS is contained in the LHS. Note that an X that commutes with all permutations is without loss of generality of the form $X = \sum_\pi \pi Y \pi^\dagger$. Inserting for Y the basis element $E_{i_1} \otimes \dots \otimes E_{i_n}$ we find that a basis for the RHS is given by $X = E_{i_1} \otimes \dots \otimes E_{i_n} +$ permutations. This element can be written as

$$\left(\prod_k \frac{\partial}{\partial t_k} \right) \left(\sum_j t_j E_{i_j} \right)^{\otimes n} \Big|_{t_1 = \dots = t_d = 0}.$$

This proves the claim since partial derivatives are defined as

$$\frac{\partial}{\partial t} (\tilde{E} + tE)^{\otimes n} \big|_{t=0} = \lim_{t \to 0} \frac{(\tilde{E} + tE)^{\otimes n} - (\tilde{E})^{\otimes n}}{t}.$$

This naturally extends to multiple derivatives.[8] The RHS is clearly the limit of a linear combination of tensor powers and thus contained in the LHS. □

[8]

$$\begin{aligned} \frac{\partial}{\partial t_2} \frac{\partial}{\partial t_1} (\tilde{\tilde{E}} + t_1 E_2 + t_1 E_1)^{\otimes n} \big|_{t=0} &= \frac{\partial}{\partial t_2} \lim_{t_1 \to 0} \frac{(\tilde{\tilde{E}} + t_2 E_2 + t_1 E_1)^{\otimes n} - (\tilde{\tilde{E}} + t_2 E_2)^{\otimes n}}{t_1} \\ &= \lim_{t_1, t_2 \to 0} \frac{(\tilde{\tilde{E}} + t_2 E_2 + t_1 E_1)^{\otimes n} - (\tilde{\tilde{E}} + t_2 E_2)^{\otimes n} - (\tilde{\tilde{E}} + t_1 E_1)^{\otimes n} + (\tilde{\tilde{E}})^{\otimes n}}{t_1 t_2} \end{aligned}$$

In order to see how irreducibility of $[k]$ follows, first note that the representation of $SU(2)$ on the tensor space extends to a representation of the matrix algebra of two-by-two complex matrices, keeping the decomposition into irreducibles intact. Now assume by contradiction that $[k]$ was reducible. Then $[k] \cong [\alpha] \oplus [\beta]$ for some nonvanishing representations $[\alpha]$ and $[\beta]$ of S_n. Then, the projector $\mathbf{1}_{V_k} \otimes P_\alpha$ onto $V_k \otimes [\alpha]$ would certainly commute with all permutations and hence be of the form $\sum_i A_i^{\otimes n}$ for some A_i (the sum is finite wlog). But by the decomposition of $V_1^{\otimes n}$ into irreducible representations, we find $\sum_i A_i^{\otimes n} = \sum_i \bigoplus_k \mathcal{V}_k(A_i) \otimes \mathbf{1}_{[k]}$ and since the total support is constrained to $V_k \otimes [k]$ and since on V_k, the operator is proportional to the identity,

$$\sum_i A_i^{\otimes n} = \sum_i \mathcal{V}_k(A_i) \otimes \mathbf{1}_{[k]} = const. \mathbf{1}_{V_k} \otimes \mathbf{1}_{[k]}.$$

This shows that $[\beta]$ must be zero-dimensional and thus $[k]$ be irreducible. In summary we find the statement of *Schur-Weyl duality*: We have the following decomposition

$$(\mathbb{C}^2)^{\otimes n} \cong \bigoplus_{n:k \mod 2 = n \mod 2} V_k \otimes [k].$$

where S_n acts irreducibly on $[k] \cong \mathbb{C}^{m_k^n}$ and $SU(2)$ acts irreducibly on V_k. If you have studied the representations of the symmetric group in another course, you might have come across that they are labelled by Young diagrams with n boxes, arranged in rows of decreasing length. Here, $[k]$ corresponds to the two-row Young diagram with $\frac{n+k}{2}$ boxes in the first, $\frac{n-k}{2}$ in the second row.

5.7 A basis for Schur-Weyl duality

By implementing the Schur transform above, we can express the basis elements $|k, l, p\rangle$ in terms of the tensor product basis, since $V_k \otimes [k] \subset \mathbb{C}^{2^{\otimes n}}$. It turns out that this is not so easy - just remember how complicated the formula for the number of paths m_k^n is! Let us therefore focus on constructing the $|k, l, p\rangle$ for the *easy* path $\tilde{p} = \tilde{p}_1 \tilde{p}_2 \cdots \tilde{p}_{n-1} = \underbrace{-+-+-\cdots+-}_{n-k-1}\underbrace{++\cdots+}_{k}$. The first minus sign means that we are in the subpace V_0 of the first two tensor factors $V_1 \otimes V_1$, i.e. the state on the first two tensor factors is the singlet $\frac{1}{\sqrt{2}}|01 - 10\rangle$. The next bit in the path is $\tilde{p}_2 = +$. This means that we tensor V_1 to V_0 and obtain $V_0 \otimes V_1$ with basis $\frac{1}{\sqrt{2}}|01 - 10\rangle \otimes |1, s\rangle$, $s \in \{0, 1\}$. (Note that $\tilde{p}_2 = -$ was not an option since we cannot decend from the trivial representation). The next bit is $\tilde{p}_3 = -$ which means that we once again obtain a singlet and have $V_0 \otimes V_0 \subset V_1 \otimes V_1 \otimes V_1 \otimes V_1$ spanned by $(\frac{1}{\sqrt{2}}|01 - 10\rangle) \otimes (\frac{1}{\sqrt{2}}|01 - 10\rangle)$. Continuing this procedure we find $(n-k)/2$ singlets tensored together $\left(\frac{1}{\sqrt{2}}|01 - 10\rangle\right)^{\otimes \frac{n-k}{2}}$. V_k is then constructed as a subrepresentation of the remaining $V_1^{\otimes k}$. The vectors

$|k,l\rangle$ that span this representation are explicitly given by

$$|k,l\rangle = \frac{1}{\sqrt{\binom{k}{l}}}\left(|\underbrace{11\dots1}_{l}\underbrace{00\dots0}_{k-l}\rangle + \text{permutations}\right).$$

This shows

$$|k,l,\tilde{p}\rangle = \left(\frac{1}{\sqrt{2}}|01-10\rangle\right)^{\otimes\frac{n-k}{2}} \otimes \operatorname{span}\{|k,l\rangle : 0 \le l \le k\}.$$

There are now two ways of constructing all the other vectors $|k,l,p\rangle$.

- Follow different paths just as we have done above. This will yield vectors orthogonal to the constructed ones, but is a little more tricky, since we don't fall back to the trivial representation as was the case for the *easy* path $\tilde{p}$.

- Apply the permutation group to the vectors $|k,l,\tilde{p}\rangle$ that we have constructed. Since the action of the permutation group commutes with the action of $SU(2)$, this will not change k nor l, but will result in superpositions over path labels: $\pi\,|k,l,p\rangle = \sum_{p'} c_{k,l,p,p'}\,|k,l,p'\rangle$ for numbers $c_{k,l,p,p'}$ and $\pi \in S_n$. Be aware that the so constructed vectors are in general not orthogonal.[9]

Other (orthogonal) copies of this representation can be constructed by following different *paths*. Other (not necessarily orthogonal copies) can be obtained by applying the permutation group to the just constructed representation.

5.8 Measurement of the Total Spin

The measurement of the total spin simply corresponds to an observable that has as its eigenspaces the projectors onto the isotypic components of V_k in $V_1^{\otimes n}$. We have seen how one can construct them explicitly above. Here we want to give a handy formula for such an observable. In order to do so note that the representation of $SU(2)$ on the tensor space induces the following representation of $su(2)$ by derivation:

$$su(2) \ni a \mapsto a \otimes \mathbf{1} \otimes \mathbf{1} \otimes \dots \otimes \mathbf{1} + \mathbf{1} \otimes a \otimes \mathbf{1} \otimes \dots \otimes \mathbf{1} + \dots + \mathbf{1} \otimes \mathbf{1} \otimes \dots \otimes \mathbf{1} \otimes a.$$

Hence, the total spin (or Casimir operator) for this representation is given by $K^2 := \vec{K}.\vec{K} = \sum_i K_i K_i$, where $\vec{K} := (K_1, K_2, K_3)$ and

$$K_i = \sigma_i \otimes \mathbf{1} \otimes \mathbf{1} \otimes \dots \otimes \mathbf{1} + \mathbf{1} \otimes \sigma_i \otimes \mathbf{1} \otimes \dots \otimes \mathbf{1} + \dots + \mathbf{1} \otimes \mathbf{1} \otimes \dots \otimes \mathbf{1} \otimes \sigma_i. \tag{5}$$

[9] As an example, note that application of a transposition of tensor factors one and two does not result in a new copy of V_k, but that transposing tensor factors two and three results in a new (non-orthogonal) copy of V_k since $\frac{1}{16}|\,\langle 01-10|_{13}\,\langle 01-10|_{24}\,|01-10\rangle_{12}\,|01-10\rangle_{34}\,|^2 \neq 1$; subscripts indicate the tensor factor.

In order to see how the Casimir acts on the space, let us decompose the K_i into their irreducible components: $K_i = \sum_k V_k(\sigma_i) \otimes \mathbf{1}_{[k]}$. Using (2) we find

$$\begin{aligned}
K^2 &= \sum_i \sum_k v_k(\sigma_i) v_k(\sigma_i) \otimes \mathbf{1}_{[k]} \\
&= \sum_k \left(\sum_i v_k(\sigma_i) v_k(\sigma_i) \right) \otimes \mathbf{1}_{[k]} \\
&= \sum_k k(k+2) P_k,
\end{aligned}$$

where P_k is the projector onto $V_k \otimes [k] \subset (\mathbb{C}^2)^{\otimes n}$. Hence we have a formula, given through (2) for an observable that measures the total spin in tensor product space.

For now, we have all mathematical tools together and turn our attention to some quantum information theory.

6 Entanglement Distillation

6.1 Formal setup

Assume two distant parties Alice and Bob share n copies of the state $|\psi\rangle_{AB}$, i.e the state $|\psi\rangle_{AB}^{\otimes n}$, which they would like to convert via LOCC, that is, local measurements and classical communication into $m \equiv m(n)$ ebits, i.e. into $|\phi\rangle_{AB}^{\otimes m}$, where $|\phi\rangle_{AB} = \frac{1}{\sqrt{2}} |00 + 11\rangle_{AB}$. We say that $R \in [0, \infty]$ is an achievable rate if for all n there exists an LOCC protocol with input $|\psi\rangle_{AB}^{\otimes n}$ and output ρ_{AB}^n such that

$$\lim_{n \to \infty} \frac{m}{n} = R \tag{6}$$

satisfying

$$\lim_{n \to \infty} \langle \phi|_{AB}^{\otimes m} \rho_{AB}^n |\phi\rangle_{AB}^{\otimes m} = 1. \tag{7}$$

We are then interested in the maximal achievable rate

$$R_{max} := \sup_R \{R \text{ achievable}\}.$$

Luckily it will turn out that the protocol achieving R_{max} not require any classical communication. Furthermore, a condition somewhat stronger than (7) is achieved: First note that

$$|\phi_{2^m}\rangle := \frac{1}{\sqrt{2^m}} \sum_{i=1}^{2^m} |i\rangle |i\rangle ,$$

and we can therefore equivalently ask Alice and Bob to distill this *maximally entangled state of rank* 2^m or even of a rank that is not a power of 2, since any such state can be converted by LOCC to a state with a rank equal to the next smaller power of two (majorisation criterion [7]). Also our output will be given by exact maximally entangled states, but (for fixed n) with a rank given by a random variable $\log m_k^n$ sharply peaked around nR. The output state

$$\tilde{\rho}_{AB}^n := \sum_k p_k |k\rangle\langle k| \otimes |\phi_{m_k^n}\rangle\langle\phi_{m_k^n}|$$

can thus easily be converted into

$$\rho_{AB}^n := |\phi_{2^{nR-O(\sqrt{n})}}\rangle\langle\phi_{2^{nR-O(\sqrt{n})}}|$$

for which it is clear that both (6) and (7) are satisfied.

6.2 The protocol

The protocol for entanglement distillation is as follows:

- Alice and Bob measure the total spin of the system and thus hold a state in $V_{k,A} \otimes V_{k,B} \otimes [k]_A \otimes [k]_B$.
- They trace out $V_{k,A}$ and $V_{k,B}$, respectively. The remaining state in $[k]_A \otimes [k]_B$ is of the form $\left|\phi_{m_k^n}\right\rangle$.

It remains to verify the claims made in the protocol and to analyse the performance. Let us start with the former. The state of Alice and Bob is of the form $|\psi\rangle^{\otimes n}$ and hence it is invariant under the action of the permutation group. Since the permutation group acts only on the parts $[k]_A \otimes [k']_B$ in

$$\bigoplus_{k,k'} V_{k,A} \otimes V_{k',B} \otimes [k]_A \otimes [k']_B,$$

and since our state is invariant under the permutation group, it must be contained in

$$\bigoplus_{k,k'} V_{k,A} \otimes V_{k',B} \otimes ([k]_A \otimes [k']_B)^{S_n},$$

where the subscript denotes the invariants in this subspace. Since an element in $([k]_A \otimes [k']_B)^{S_n}$ can be regarded as an S_n invariant map from $[k]_A$ to $[k']_B$ and since the action on $[k]$ is irreducible, this map can only be proportional to the identity according to Schur's lemma, Lemma 2. Equivalently, the state is proportional to the maximally entangled state. Hence

$$|\psi\rangle_{AB}^{\otimes n} = \sum_k c_k \left|\psi_k\right\rangle_{V_k \otimes V_k} \otimes \left|\phi_{m_k^n}\right\rangle_{[k]\otimes[k]}.$$

It remains to compute the probabilities $|c_k|^2$.

$$\begin{aligned} |c_k|^2 &= \mathrm{tr} P_{k,A} \otimes P_{k,B} |\psi\rangle\langle\psi|_{AB}^{\otimes n} \\ &= \mathrm{tr} P_{k,A} \otimes \mathbf{1}_{B^n} |\psi\rangle\langle\psi|_{AB}^{\otimes n} \\ &= \mathrm{tr} P_{k,A} \rho_A^{\otimes n}, \end{aligned}$$

where $\rho_A = \mathrm{tr}_B|\psi\rangle\langle\psi|_{AB}$ with eigenvalues $(r, 1-r)$ which we assume to satisfy $0 < r < \frac{1}{2}$ (the cases $r \in \{0, \frac{1}{2}\}$ can easily be verified separately). The second inequality holds since the state has no component on $[k] \otimes [k']$ for $k \neq k'$.

$$\begin{aligned}
|c_k|^2 &= \mathrm{tr} P_k \rho_A^{\otimes n} \\
&= m_k^n \mathrm{tr}|\phi\rangle\langle\phi|^{\otimes \frac{n-k}{2}} \otimes \left(\sum_l |k,l\rangle\langle k,l|\right) \rho_A^{\otimes n} \\
&= m_k^n \left(r(1-r)\right)^{\frac{n-k}{2}} \left(\sum_{l=0}^{k} r^l (1-r)^{k-l}\right) \qquad (8) \\
&= m_k^n \left(r(1-r)\right)^{\frac{n-k}{2}} (1-r)^k \sum_{l=0}^{k} \left(\frac{r}{1-r}\right)^l \\
&= m_k^n r^{\frac{n-k}{2}} (1-r)^{\frac{n+k}{2}} \left(\frac{1-(\frac{r}{1-r})^{k+1}}{1-\frac{r}{1-r}}\right) \\
&\approx m_k^n r^{\frac{n-k}{2}} (1-r)^{\frac{n+k}{2}} \left(\frac{1-r}{1-2r}\right) \\
&= \binom{n}{\frac{n-k}{2}} r^{\frac{n-k}{2}} (1-r)^{\frac{n+k}{2}} \left(\frac{2k+2}{n+k+2} \frac{1-r}{1-2r}\right) \qquad (9) \\
&\approx const. \binom{n}{\frac{n-k}{2}} r^{\frac{n-k}{2}} (1-r)^{\frac{n+k}{2}}
\end{aligned}$$

Line (8) follows since we choose the basis $\{|0\rangle, |1\rangle\}$ as the eigenbasis of ρ_A. In order to obtain Line (9) we have inserted Eq. (4). The last line is true for k which are linear in n – this we can assume as for smaller k, $|c_k|^2$ decreases exponentially.

By the law of large numbers (see also Figure 2), the $|c_k|^2$ is highly peaked for $k \approx n(2r-1)$. Using the estimates for m_k^n we find

$$\lim_{n\to\infty} \mathbf{E}\left(\frac{\log m_k^n}{n}\right) = h(r).$$

In other words $h(r) \leq R_{max}$. That also $h(r) \geq R_{max}$ will be shown in the next two sections.

6.3 The converse: entanglement dilution

We are now considering the task of entanglement dilution. Here, it is the goal to construct as many copies of $|\psi\rangle_{AB}$ as possible per ebit with local operations and classical communication only. Formally, an achievable rate R for entanglement dilution satisfies

$$\lim_{n\to\infty} \frac{n}{m} = R.$$

and

$$\lim_{n\to\infty} \langle\psi|_{AB}^{\otimes n} \rho_{AB}^n |\psi\rangle_{AB}^{\otimes n} = 1,$$

where ρ^n_{AB} is the output of the protocol that takes as input m ebits. In Figure 3, the following protocol is illustrated which achieves a rate $R = \frac{1}{h(r)+\epsilon}$ (for all $\epsilon > 0$).

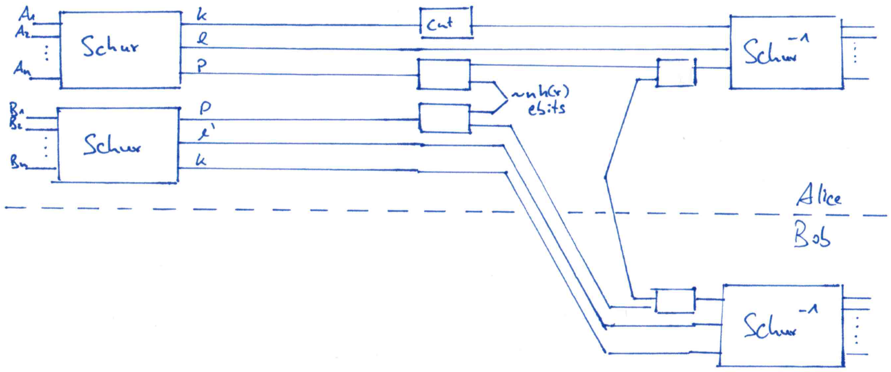


Figure 3: Circuit for entanglement dilution: Alice creates locally the state $|\psi\rangle^{\otimes n}_{AB}$ whose B part is *merged* to Bob by sending $O(\sqrt{n}\log n)$ bits of (classical or quantum) communication and the use of $\approx nh(r)$ ebits.

- Alice locally creates the state $|\psi\rangle^{\otimes n}_{AB}$ and applies the Schur transform to the A and B parts separately.
- Alice measures the projector $P_\epsilon = \sum_{k\in n[2r-1-2\epsilon, 2r-1+2\epsilon]} |k\rangle\langle k|$ and its complement and continues if she obtains the first outcome (the probability for this event approaches one for large n, see Figure 2).
- If Alice continues then there are between $n(h(r) - O(\epsilon\log\epsilon)) + O(\log n)$ and $n(h(r) + O(\epsilon\log\epsilon))$ path ebits.[10]
- The $n(h(r) - O(\epsilon\log\epsilon)) + O(\log n)$ ebits will be exchanged against ebits shared with Bob, and all the remaining outputs from the Schur transform on the B systems are teleported to Bob.
- Alice and Bob both apply the inverse Schur transform and obtain the state ρ^n_{AB}. [11]

[10] This follows from the inequality $|h(x-\epsilon) - h(x)| \leq h(\epsilon)$, whose quantum generalisation is known as Fannes' inequality.

[11] The protocol has consumed $n(h(r) - O(\epsilon\log\epsilon)) + O(\log n)$ ebits for the exchange plus $O(n\epsilon\log\epsilon)$ ebits for the teleportation. The teleportation also requires $O(n\epsilon\log\epsilon)$ bits of classical communication. Since the peak in Figure 2 is $O(\frac{1}{\sqrt{n}})$ broad, we can choose $\epsilon = O(\frac{1}{\sqrt{n}})$ and hence $O(\sqrt{n}\log n)$ bits of communication are sufficient. This can easily be improved to $O(\sqrt{n})$ which is also optimal [4]. This is to be contrasted to entanglement distillation where no communication was needed.

6.4 Optimality

It is the goal of this section to show that the entanglement distillation rate of $h(r)$ is optimal. The argument is simple: Assume we could perform entanglement distillation with rate $R_{max} > h(r)$. Then we could first convert n ebits into $\approx \frac{1}{h(r)} n$ copies of $|\psi\rangle_{AB}$ and subsequently distill $\approx \frac{R_{max}}{h(r)} n$ ebits.

$$|\phi\rangle^{\otimes n} \xrightarrow{\approx} |\psi\rangle^{\otimes \frac{1}{h(r)} n} \xrightarrow{\approx} |\phi\rangle^{\otimes \frac{R_{max}}{h(r)} n}$$

So, if $R_{max} > h(r)$ we could convert n ebits into $\approx cn$ ebits, where $c > 1$!

Note that this would imply in particular an increase in the number of nonzero Schmidt coefficients from the initial to the final state.[12] But this is not possible, since any local measurement cannot increase the number of nonzero Schmidt coefficients of the state (and any allowed operation consists of local measurements and the transmission of classical information).

In summary, we have shown that $R \leq h(r)$, which proves that the our protocol for entanglement distillation is optimal. By the same token, the achievability of the entanglement distillation rate of $h(r)$ implies the optimality of the entanglement dilution rate of $\frac{1}{h(r)}$.

7 Permutational Quantum Computer

7.1 Introduction

In the beginning of this course, we have introduced spin as a property of a particle and we have argued that this particle has to transform according to representations of the group $SU(2)$, the covering group of the rotation group in three dimensional space, $SO(3)$. We have then taken many particles carrying spin (in fact, we restricted our attention to the representation V_1) and studied how $SU(2)$ transforms the state of those particles. We have also studied the effect of permuting the particles. It turned out that both actions where maximal commutants of each other and this led us to introduce a convenient basis in which to express this action. We have also seen that the representations of the symmetric group are very large, meaning that the Hilbert space (where the path information is stored) on which they act, grows exponentially with the number of particles. It is therefore very natural to ask whether we can perform a computation on the basis states by permuting the particles. This is the subject of this chapter. The discussion is inspired by the exposition in [6].

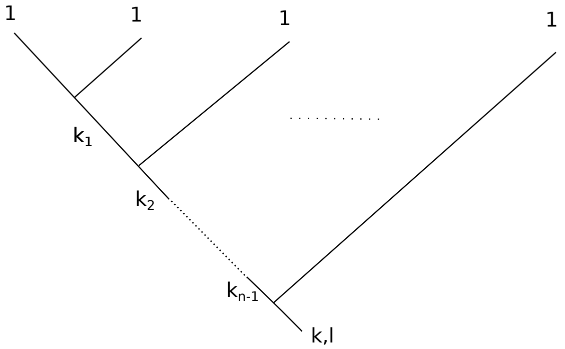
From a physical point of view it is a very interesting idea to perform computation by permutation, since the quantum gates (the transpositions) are robust in the sense that it does not matter, which exact route the particles follow when being exchanged but only *that* they have been exchanged. This would eradicate a typical problem with quantum computers, namely that gates have to be implemented with high precision.

[12] n ebits have exactly 2^n nonzero Schmidt coefficients.

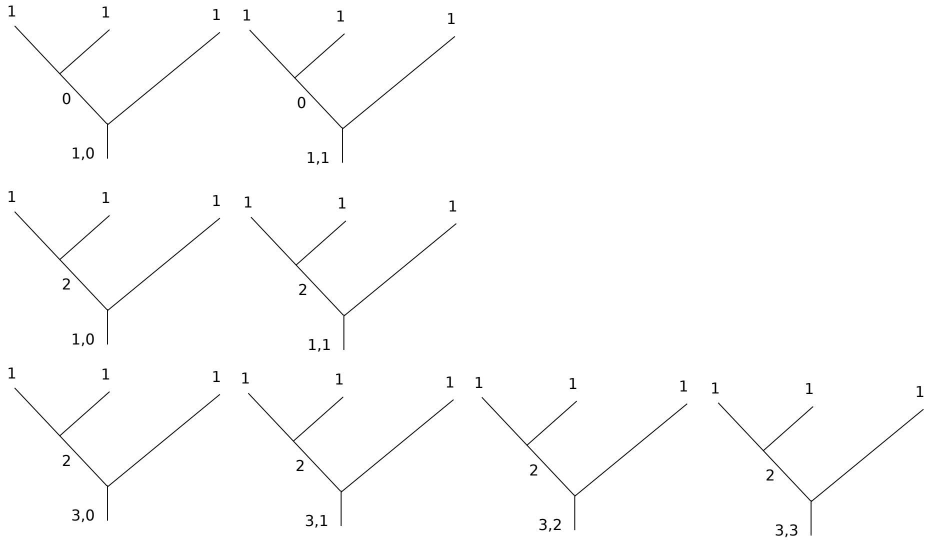
7.2 Tree diagrams

We have already seen that the handling of the states $|k, l, p\rangle$ is highly non-trivial and that acting with the permutation group does not make it better. The first thing we therefore do, is introduce a diagrammatic method that lets us handle these states a little better.

We will represent the state $|k, l, p\rangle$ by the following diagram:



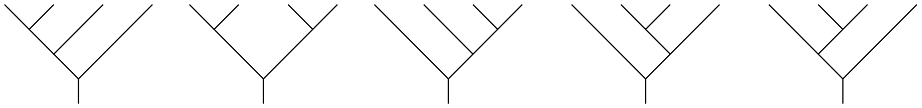
Here, time flows downwards and space to the right. Each wire represents a particle, that is an irreducible representation. When two wires come together – this is known as fusion – this corresponds to an application of the Clebsch-Gordan transform. Instead of recording the path label p_i (that is whether the outcome was higher or lower), we record the representation k_i that resulted from the fusion. We indicate the weight of the vector at the bottom of the diagram. Each distinct label corresponds to a distinct orthonormal state. The following is a list of the eight orthonormal states that are obtained by fusing three particles:



Since the Clebsch-Gordan transform as well as the action of the permutation group cannot change the weight, we will suppress this label in our notation and think of it as being fixed. More formally, we can eliminate the weight label on our tree if we regard the tree as an $SU(2)$-invariant homomorphism from $V_1^{\otimes n}$ to V_k (i.e. the projector onto a specific copy of V_k contained in $V_1^{\otimes n}$) that is an element in $\mathrm{Hom}(V_1^{\otimes n}, V_k)$.

7.3 Recoupling

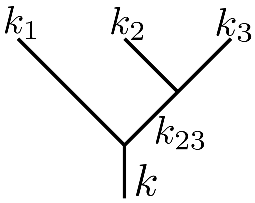
Note that the notation clearly indicates that we started fusing the particles from the left. Of course we might also consider other orders in which to fuse the particles. In the example of four particles we have the following possibilities:



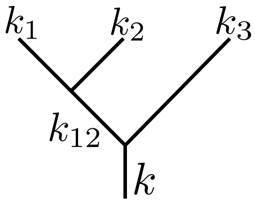
The first and familiar fusion order is known as *left standard*, the third as the *right standard*. We want to think of all the lose ends having fixed irreducible representations attached to them. Above we had always considered the case, where the top row has all labels equals to 1 and the bottom label equals k (but this restriction is not really necessary). When varying over the path labels or internal fusion labels, we obtain an orthonormal basis for the space $\mathrm{Hom}(V_1^{\otimes n}, V_k)$. Each choice of fusion order results in a different basis for this same space.

But how can we transform between two such bases? Since we know how we obtained them the answer is easy: starting from one basis we first have to undo the Clebsch-Gordan transforms according to the fusion order of that basis so that we land in the computational basis. Then we perform the Clebsch-Gordan transforms corresponding to the new bases. All in all we obtain an expression of one basis in terms of the other.

But let us do this basis transform slowly at the example of fusing three particles. Here there are only two bases, the left standard basis and the right standard basis and we wish to find the transformation between them, known as the F-matrix:



$$= \sum_{k_{12}} \begin{bmatrix} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{bmatrix}$$



The coefficients are known as recoupling or (Wigner) $6j$-coefficients (be aware of the different normalisations that are in use). Before we state the recoupling coefficients in terms of the Clebsch-Gordan coefficients (which are also known as the (Wigner) $3j$-coefficients), let us introduce a notation for the latter:[13]

$$|k_1,k_2,k,l\rangle = \sum_{l_1,l_2} \left(\begin{array}{ccc} k_1 & k_2 & k \\ l_1 & l_2 & l \end{array} \right) |k_1,l_1\rangle\,|k_2,l_2\rangle \ .$$

This allows us to write

$$\begin{aligned} &|k_1,k_2,k_3,k_{12},k,l\rangle_{\text{left}} \\ &\quad = \sum_{l_1,l_2,l_3,l_{12}} \left(\begin{array}{ccc} k_1 & k_2 & k_{12} \\ l_1 & l_2 & l_{12} \end{array} \right) \left(\begin{array}{ccc} k_{12} & k_3 & k \\ l_{12} & l_3 & l \end{array} \right) |k_1,l_1\rangle\,|k_2,l_2\rangle\,|k_3,l_3\rangle \end{aligned}$$

as well as

$$\begin{aligned} &|k_1,k_2,k_3,k_{23},k,l\rangle_{\text{right}} = \\ &\quad \sum_{l_1,l_2,l_3,l_{23}} \left(\begin{array}{ccc} k_2 & k_3 & k_{23} \\ l_2 & l_3 & l_{23} \end{array} \right) \left(\begin{array}{ccc} k_1 & k_{23} & k \\ l_1 & l_{23} & l \end{array} \right) |k_1,l_1\rangle\,|k_2,l_2\rangle\,|k_3,l_3\rangle \ . \end{aligned}$$

Hence we obtain

$$\begin{aligned} \left[\begin{array}{ccc} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{array} \right] &:= \langle k_1,k_2,k_3,k_{12},k,l|_{\text{left}}\, |k_1,k_2,k_3,k_{23},k,l\rangle_{\text{right}} \\ &= \sum_{l_1,l_2,l_3,l_{12},l_{23}} \left(\begin{array}{ccc} k_1 & k_2 & k_{12} \\ l_1 & l_2 & l_{12} \end{array} \right) \left(\begin{array}{ccc} k_{12} & k_3 & k \\ l_{12} & l_3 & l \end{array} \right) \\ &\qquad \left(\begin{array}{ccc} k_2 & k_3 & k_{23} \\ l_2 & l_3 & l_{23} \end{array} \right) \left(\begin{array}{ccc} k_1 & k_{23} & k \\ l_1 & l_{23} & l \end{array} \right) , \end{aligned}$$

where we made use of the fact that all the Glebsch-Gordan coefficients are real. It is easy to see that recoupling moves are sufficient to transform any basis into any other basis. Note that for given k_1, k_2, k_3 and k, the matrix F with elements $\left[\begin{array}{ccc} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{array} \right]$ is unitary. Since the coefficients are furthermore real we find $F^{-1} = F^\dagger = F^T$ where T denotes the transpose. In other words

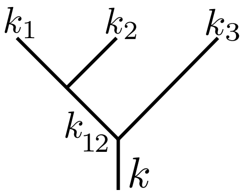
$$\sum_{k_{12}} \left[\begin{array}{ccc} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{array} \right] \left[\begin{array}{ccc} k_1 & k_2 & k_{12} \\ k_3 & k & k'_{23} \end{array} \right] = \delta_{k_{23},k'_{23}}$$

and

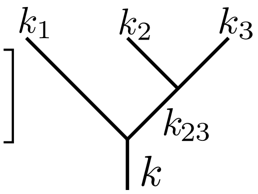
$$\sum_{k_{23}} \left[\begin{array}{ccc} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{array} \right] \left[\begin{array}{ccc} k_1 & k_2 & k'_{12} \\ k_3 & k & k_{23} \end{array} \right] = \delta_{k_{12},k'_{12}}$$

This implies

[13] In the notation of Section 5.3 we have $\langle k_1 l_1|\,\langle k_2 l_2|\,|k,l\rangle \equiv \left(\begin{array}{ccc} k_1 & k_2 & k \\ l_1 & l_2 & l \end{array} \right)$ and moreover we calculated the Glebsch-Gordan coefficients for the special case $k_2 = 1$ there.

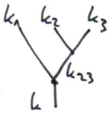


$$= \sum_{k_{23}} \begin{bmatrix} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{bmatrix}$$

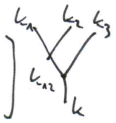


since

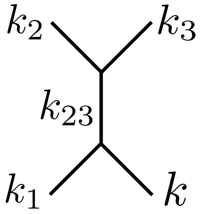
$$\sum_{k_{23}} \begin{bmatrix} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{bmatrix}$$



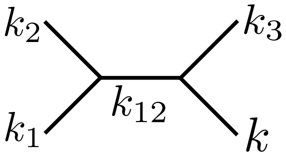
$$= \sum_{k'_{12}} \underbrace{\sum_{k_{23}} \begin{bmatrix} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{bmatrix} \begin{bmatrix} k_1 & k_2 & k'_{12} \\ k_3 & k & k_{23} \end{bmatrix}}_{\delta_{k'_{12},k_{12}}}$$



Modifying the lines in the diagram a bit, we see why this process is also known as recoupling:



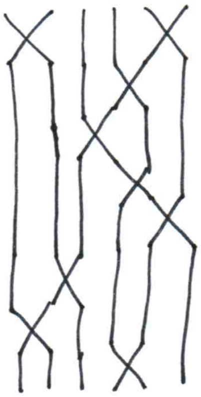
$$= \sum_{k_{12}} \begin{bmatrix} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{bmatrix}$$



7.4 Permutation

This section is guided by the question of whether the permutation of particles can be used to perform useful computation.

An entire permutation can be viewed as a permutational circuit, graphically

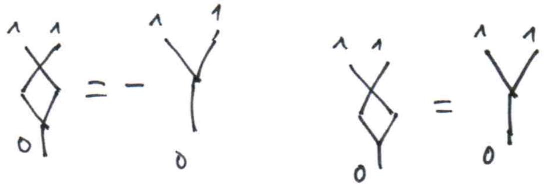


The transposition of particle i and $i+1$ is denoted by π_i and represented as

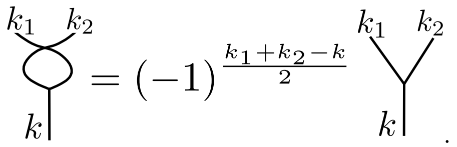
$$\pi_i \mapsto \underbrace{\mathbf{1}\otimes\mathbf{1}\otimes\cdots\otimes\mathbf{1}}_{i-1}\otimes\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\otimes\underbrace{\mathbf{1}\otimes\mathbf{1}\otimes\cdots\otimes\mathbf{1}}_{n-(i+1)}$$

when expressed in terms of the computational basis.

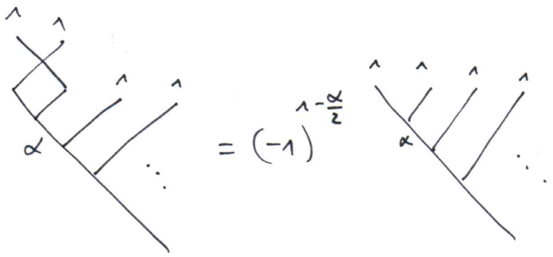
In the remainder of this section we want to express this transposition in terms of the tree bases. Let us start step by step and consider first the case where two particles fuse. When both particles are of type one and fuse to zero then they are in a singlet. An exchange of the particles will therefore result in a minus sign. When the particles fuse to a 2, they are in the triplet and an exchange does not change the state at all. Represented graphically, we summarise



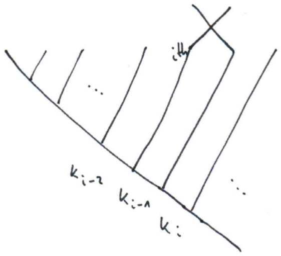
This argument can be generalised to an exchange of particles of types k_1 and k_2 that fuse to k



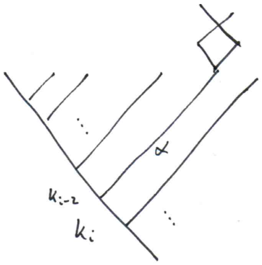
So, if we want to compute the effect of exchanging particles one and two in a basis state of the left standard basis this not a problem, we just pick up a minus sign if they fuse to 0 and a plus sign if they fuse to 1.



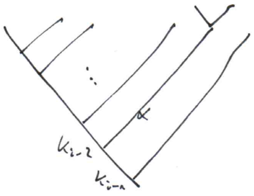
But what do we do when we want to permute any of the other particles? We first use recoupling moves in order to transform one basis into a basis where the particles fuse directly, we then apply the permutation, and then we undo the recoupling moves. Let us do this explicitly with the example of the left standard basis.



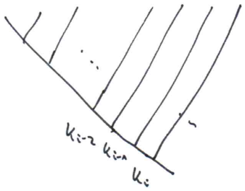
$$= \sum_\alpha \begin{bmatrix} k_{i-2} & 1 & k_{i-1} \\ 1 & k_i & \alpha \end{bmatrix}$$



$$= \sum_\alpha \begin{bmatrix} k_{i-2} & 1 & k_{i-1} \\ 1 & k_i & \alpha \end{bmatrix} (-1)^{1-\frac{\alpha}{2}}$$



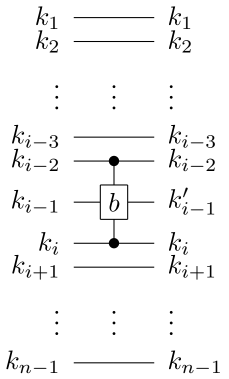
$$= \sum_\alpha \begin{bmatrix} k_{i-2} & 1 & k_{i-1} \\ 1 & k_i & \alpha \end{bmatrix} (-1)^{1-\frac{\alpha}{2}} \begin{bmatrix} k_{i-2} & 1 & k'_{i-1} \\ 1 & k_i & \alpha \end{bmatrix}$$



Note that with only one recoupling move we have obtained the desired position. Hence, the transposition π_i is represented by a single qubit unitary acting on k_i controlled by k_{i-1} and k_{i+1}:

$$\pi_i \mapsto b_i$$

where b_i is given by the circuit



where the coefficients of the matrix b are

$$\begin{aligned} b^{k_{i-2},k_i}_{k'_{i-1},k_{i-1}} &= \begin{bmatrix} k_{i-2} & 1 & k_{i-1} \\ 1 & k_i & 2 \end{bmatrix} \begin{bmatrix} k_{i-2} & 1 & k'_{i-1} \\ 1 & k_i & 2 \end{bmatrix} \\ &\quad - \begin{bmatrix} k_{i-2} & 1 & k_{i-1} \\ 1 & k_i & 0 \end{bmatrix} \begin{bmatrix} k_{i-2} & 1 & k'_{i-1} \\ 1 & k_i & 0 \end{bmatrix}. \end{aligned}$$

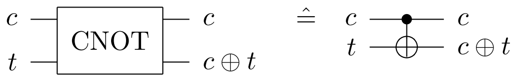
Note that each wire carries states of dimension at most n, hence our permutation acts on a n^3 dimensional system. As we will see below, such a unitary can be decomposed in poly(n) CNOT and single qubit gates. This shows that we can simulate our permutational computation model as a standard quantum circuit (with only polynomial overhead).

Two remarks are in order that show how limited the power of this computational model is. First, note that each permutation corresponds to a circuit and that each circuit corresponds to a permutation. We will see in the exercises that every permutation in S_n can be realised with only $O(n^2)$ transpositions. Therefore, arbitrary permutational circuits can be simulated with a circuit of size poly(n). Hence, in this computational model, any circuit is an efficient circuit. More aspects of the complexity theory of this model are discussed in [6]. Second, note that we cannot realise any unitary transformation (not even on a subspace) by a sequence of permutations, since there are infinitely many unitaries but only $n!$ different permutations. We therefore say that the model is not universal for quantum computation. In the following section we will make an excursion into the circuit model, in order to understand the concept of universality in more detail.

8 The Circuit Model is Universal for Quantum Computation

After having seen that the permutational model is not universal for quantum computation, it is our goal to show that the circuit model indeed is. This would provide a nice justification for the use of the circuit model in the first place, something we have already done at different place int his course.

Recall the CNOT gate



and the single qubit gates, which are gates acting on one qubit only.

Theorem 4. *The CNOT gate together with arbitrary single qubit gates are universal for quantum computation. More precisely, any $d \times d$ unitary U can be written in the form*

$$U = U_1 U_2 \cdots U_k,$$

where the U_j are one-qubit or CNOT gates and k is a finite number bounded by poly(d).

Proof. The proof consists of two steps. In a first step, we will decompose U into so-called two-level unitaries that is unitaries that affect two basis vectors only. In a second step, we will decompose those in terms of CNOT and single qubit unitaries.

We consider an arbitrary unitary

$$U = \begin{bmatrix} a & * & * & \cdots \\ b & * & * & \cdots \\ c & * & * & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Upon multiplying U from the left with the *two-level* unitary

$$U_1^\dagger := \begin{bmatrix} \frac{\bar{a}}{\sqrt{|a|^2+|b|^2}} & \frac{\bar{b}}{\sqrt{|a|^2+|b|^2}} & 0 & \cdots & 0 \\ \frac{b}{\sqrt{|a|^2+|b|^2}} & \frac{-a}{\sqrt{|a|^2+|b|^2}} & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

we find

$$U_1^\dagger U = \begin{bmatrix} a' & * & * & \cdots \\ 0 & * & * & \cdots \\ c' & * & * & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

We then multiply again from the left with

$$U_2^\dagger := \begin{bmatrix} \frac{\bar{a}'}{\sqrt{|a'|^2+|c'|^2}} & 0 & \frac{\bar{c}'}{\sqrt{|a'|^2+|c'|^2}} & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ \frac{c'}{\sqrt{|a'|^2+|c'|^2}} & 0 & \frac{-a'}{\sqrt{|a'|^2+|c'|^2}} & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

and find

$$U_2^\dagger U_1^\dagger U = \begin{bmatrix} a'' & * & * & \cdots \\ 0 & * & * & \cdots \\ 0 & * & * & \cdots \\ d'' & * & * & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

We continue this way until all but the first element in the first column vanish. Since all matrices involved are unitary, this first element must be of modulus one and the remaining elements in the first row must vanish. After adjusting the phase of the first element we find

$$\begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & * & * & \cdots \\ 0 & * & * & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

We then perform the same procedure on the smaller block and continue until we obtain the identity, i.e.

$$U_\ell^\dagger U_{\ell-1}^\dagger \cdots U_1^\dagger U = \mathbf{1},$$

which shows that U can be decomposed into $\ell \leq d(d+1)/2$ two-level unitaries:

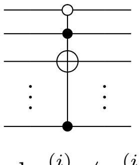
$$U = U_1 \cdots U_\ell.$$

We will now show how an arbitrary two-level unitary V can be implemented with single qubit operations and the CNOT gate. By definition, V acts non-trivially on at most two basis vectors. Without loss of generality, let it act non-trivially on exactly two vectors from the computational basis indexed by $s = s_1 \cdots s_n$ and $t = t_1 \cdots t_n$ where n is the smallest integer such that the dimension of the unitary U is smaller than or equal to 2^n. When s and t differ in only one position, the two-level unitary is a single qubit operation. It is thus our aim to transform s into a vector s' which differs only in one position with t, then apply a single qubit operation and then transform the vector back again. Let c be the number of positions in which s and t are different. By changing one after the other the bits in s that differ from the ones in t we obtain $c-1$ vectors $s = s^{(0)}, s^{(1)}, \cdots, s^{(c-1)} = s', t$, a set of vectors called a Gray code.

Example 1.

$$\begin{aligned} s = \quad s^{(0)} &= \quad 01011001001 \\ s^{(1)} &= \quad 01111001001 \\ s^{(2)} &= \quad 01110001001 \\ s^{(3)} &= \quad 01110101001 \\ t = \quad s^{(4)} &= \quad 01110101011 \end{aligned}$$

We can swap the vectors $s^{(i)}$ and $s^{(i+1)}$ with the following circuit, where we control on the bits where $s^{(i)}$ and $s^{(i+1)}$ are equal (the open bullet means conditioning on 0). The target is the index, where the two strings differ:



(here $s_1^{(i)} = 0$, $s_2^{(i)} = 1$, $s_n^{(i)} = 1$ and $s_3^{(i)} \neq s_3^{(i+1)}$). After $c-1$ such circuits we apply the single qubit gate, which is defined by the two-level unitary, on the bit where s' and t differ controlled by all other bits of s'. Subsequently, we undo the swaps.

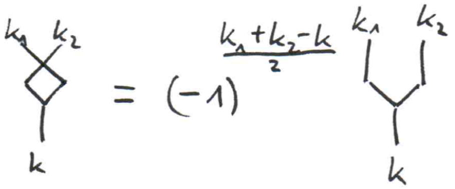
It remains to be shown that we can implement multiply controlled single qubit gates with help of single qubit gates and CNOT gates. This was done in the exercises. □

This shows that the circuit model is universal for quantum computation.

9 Topological Quantum Computer

9.1 Particles in two and three dimensions

When we exchange two particles twice in clockwise direction this corresponds to winding one particle around the other. As a physical operation this operation should have a unitary matrix as its mathematical equivalent. In three space dimensions, however, the path winding one particle around the other is easily seen to be contractible to the trivial path that leaves both particles where they are. This implies that the unitary matrix corresponding to a double particle exchange must equal the identity matrix and this again shows that the unitary matrix representing particle exchange can only have eigenvalues one and minus one. And indeed this is what we had found above



This argument can be related to a rotation around itself, something we have discussed in the beginning of the course in the context of the double cover $SU(2)$ of the rotation group $SO(3)$, but we will not discuss this connection in more detail in this course. The interested reader is referred to John Preskill's lecture notes and his remarks about the relation between spin and statistics.

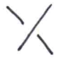
Interestingly, particle exchange is different in two dimensions. Here, a path of one particle around the other cannot be deformed into the trivial path and hence does not have to be represented by the identity matrix. In consequence, particle exchange in a two-dimensional world may be represented by a unitary matrix that does not only have eigenvalues one or minus one (or equivalently exchange phases of 0 or π) but may have *any* phase. In analogy to bosons and fermions, such particles are called anyons.

In our three-dimensional world we cannot hope to have elementary particles that behave like anyons, even if we confine them two a two-dimensional surface since we could always remove the confinement. In certain materials, however, we may hope to see quasi-particles (or excitations) that behave like anyons. There are several candidate materials having anyonic excitations, most famously the two-dimensional electron gases exhibiting the fractional quantum Hall effect. Mathematically easier to understand are the anyonic excitations in several lattice models, mostly generalisations of Kitaev's toric code.

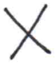
Rather than discussing a specific model, we will introduce a general framework in which *all* such models can be treated, but before let us take a look at the braid group, the group governing the exchange of anyons.

9.2 The braid group

So what could we do if we had anyons at our hand? We have argued above that exchanging anyons twice in the same direction, for instance clockwise, is not the same as doing nothing. In other words, exchanging particles clockwise or counterclockwise may make a difference in two spatial dimensions. We may therefore represent a clockwise exchange of particles by the following diagram



which replaces our particle exchange in the three-dimensional world where clockwise and counterclockwise exchange were identical.



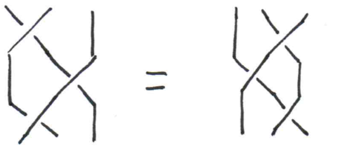
In three dimensions, the exchange of n particles was governed by the symmetric group S_n acting on n strands - in two dimensions the relevant group is the braid group B_n. Let τ_i, $i = \{1, \ldots, n-1\}$ be the generators of the braid group on n strands exchanging strand i and $i+1$ in clockwise manner. The braid group is then characterised by the following set of algebraic relations: When two exchanges act on entirely different strands then

$$\tau_i \tau_j = \tau_j \tau_i \qquad |i-j| \geq 2 \tag{10}$$

whereas when they have a strand in common the following relation holds

$$\tau_i \tau_{i+1} \tau_i = \tau_{i+1} \tau_i \tau_{i+1} \tag{11}$$

Represented graphically for three-strand braids it reads

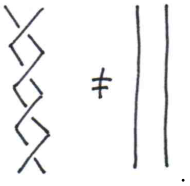


Note that the symmetric group has one relation in addition, namely

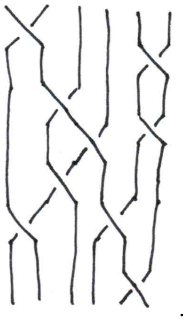
$$(\tau_i)^2 = e \tag{12}$$

where e is the identity element.

Since the symmetric group S_n has only $n!$ different elements we had seen above that exchanging particles in three dimensions cannot lead to a universal model for quantum computation. This argument does not hold anymore for particles in two dimensions since we can easily see that the braid group B_n has an infinite number of elements. Even for two strands every additional exchange of the strands (in the same direction) results in a new braid.



So there is the hope that we may perform universal quantum computation (or at least a very good approximation of it, since with a discrete number of braids we can certainly not get an arbitrary unitary exactly) by braiding particles with a circuit looking like this:



But in order to have a quantum mechanical particle model, it is not sufficient to play around with strands. We need a representation of the braid group. Let us recall how we obtained the representations of the symmetric group in the attempt to generalise this approach. Here, each strand was represented by a vector space $V \cong \mathbb{C}^2$ (we disregard that this space was endowed with an action of $SU(2)$) and the symmetric group was acting as

$$S_n \ni \pi_i \mapsto \underbrace{\mathbf{1} \otimes \mathbf{1} \otimes \cdots \otimes \mathbf{1}}_{i-1} \otimes \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \otimes \underbrace{\mathbf{1} \otimes \mathbf{1} \otimes \cdots \otimes \mathbf{1}}_{n-(i+1)}$$

when expressed in terms of the computational basis. This action may easily be generalised to arbitrary local dimensions. Since it is a representation of S_n, the matrices fulfill equations (10) (11) and (12). It is then natural to ask if we can find a modification of

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

that violates (12), but still satisfies (10) and (11)? In other words, can we find a non-trivial representation of the braid group by deforming particle exchange? Formally, we are looking for an element $b : V \otimes V \to V \otimes V$ that satisfies the following equation, known as the Yang-Baxter equation

$$b_{12}b_{23}b_{12} = b_{23}b_{12}b_{23}. \tag{13}$$

Each side of the equation acts on $V \otimes V \otimes V$ and the subscript of b indicates on which two tensor factors b acts.[14] It is indeed possible to find representations of the braid group this way – and it is also possible to generalise Schur-Weyl duality to the braid group and its dual, but unfortunately this is not so easy. We therefore choose a different route: we will build anyon models directly. Luckily we are well-prepared for this endeavour!

[14] Sometimes, the following different equation is called the Yang-Baxter equation

$$R_{23}R_{13}R_{12} = R_{12}R_{13}R_{23}, \tag{14}$$

where $R : V \otimes V$. The element b and R are then related by a permutation:

$$b = \pi R$$

where π is the exchange operator on $V \otimes V$, i.e.

$$\pi = \sum_{k,l} |l\rangle \langle k| \otimes |k\rangle \langle l|$$

It remains to verify that (13) is equivalent to (14)

$$b_{12}b_{23}b_{12} = b_{23}b_{12}b_{23}$$

The LHS equals

$$\begin{aligned} b_{12}b_{23}b_{12} &= \pi_{12}R_{12}\pi_{23}R_{23}\pi_{12}R_{12} \\ &= \pi_{12}R_{12}\pi_{23}\pi_{12}R_{13}R_{12} \\ &= \pi_{12}\pi_{23}R_{13}\pi_{12}R_{13}R_{12} \\ &= \pi_{12}\pi_{23}\pi_{12}R_{23}R_{13}R_{12} \\ &= \pi_{13}R_{23}R_{13}R_{12}. \end{aligned}$$

The RHS equals

$$\begin{aligned} b_{23}b_{12}b_{23} &= \pi_{23}R_{23}\pi_{12}R_{12}\pi_{23}R_{23} \\ &= \pi_{23}R_{23}\pi_{12}\pi_{23}R_{13}R_{23} \\ &= \pi_{23}\pi_{12}R_{13}\pi_{23}R_{13}R_{23} \\ &= \pi_{23}\pi_{12}\pi_{23}R_{23}R_{13}R_{23} \\ &= \pi_{13}R_{23}R_{13}R_{23} \end{aligned}$$

and hence the statement is equivalent to (14).

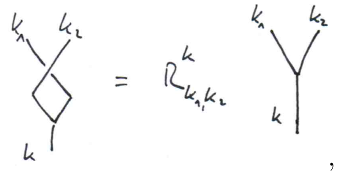
9.3 Anyon models

An anyon model, sometimes known as a braided tensor category, is given by the following set of data:

- *Particle types* are labeled by elements from a discrete (mostly finite) set, for instance $\{0, 1, 2, \ldots\}$.
- *Fusion rules* tell us the possibilities of the outcomes when two particles, k_1 and k_2, are fused.

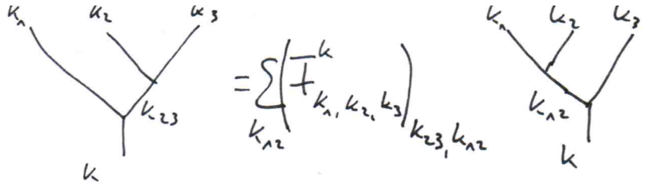
  $$k_1 \times k_2 = \sum_k N^k_{k_1,k_2} k$$

  where $N^k_{k_1,k_2}$ is the number of different ways in which two particles fuse to a particular third particle. Not to clutter our notation, we will only consider fusion rules where there is at most one way, i.e. $N^k_{k_1,k_2} \in \{0, 1\}$.
- *Braiding rules* tell us what happens when particles are being exchanged. Braiding of two particles does not affect the particle to which they fuse, hence

  ,

  where $R^k_{k_1,k_2} = e^{i\Theta^k_{k_1,k_2}}$ is a phase factor.
- The *F-matrix* relates the different orders in which particles can be fused.

  
  k_1
  k_2
  k_3
  k_{23}
  k
  $= \sum_{k_{12}} \left(F^k_{k_1,k_2,k_3}\right)_{k_{23},k_{12}}$
  k_1
  k_2
  k_3
  k_{12}
  k

Of course there are some consistency requirements that this data set has to satisfy in order to be an anyon model. But let us first look at our $SU(2)$ example:

Example 2. *$SU(2)$*

- *Particle types are the different values of spin, in our case labelled by a non-negative integer,* $0, 1, 2,$ *etc.*

- *The fusion rule is*

$$k_1 \times k_2 = \sum_{k=|k_1-k_2|:k \ mod\ 2=k_1+k_2\ mod\ 2}^{k_1+k_2}$$

  understood as abbreviating $V_{k_1} \otimes V_{k_2} = \bigoplus V_k$*. In general, there must not be any vector spaces associated with a certain particle type and the fusion rule does have to arise as a representation of a group.*

- *The braiding rules correspond to definite phases when two particles that fuse to a certain particle are being exchanged. In our case these complex numbers are* $R^k_{k_1,k_2} = (-1)^{\frac{k_1+k_2-k}{2}}$*. In general these may be any phases.*

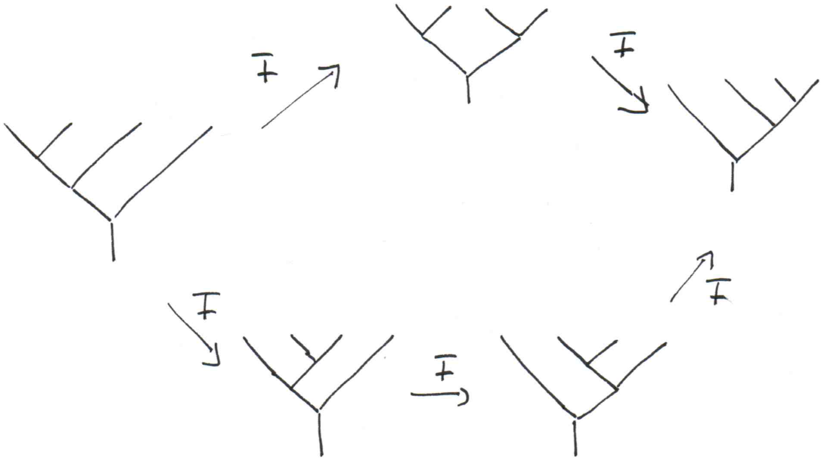
- *The* $6j$*-coefficients that we have computed from the Clebsch-Gordan coefficients are the entries of the* F*-matrix.*

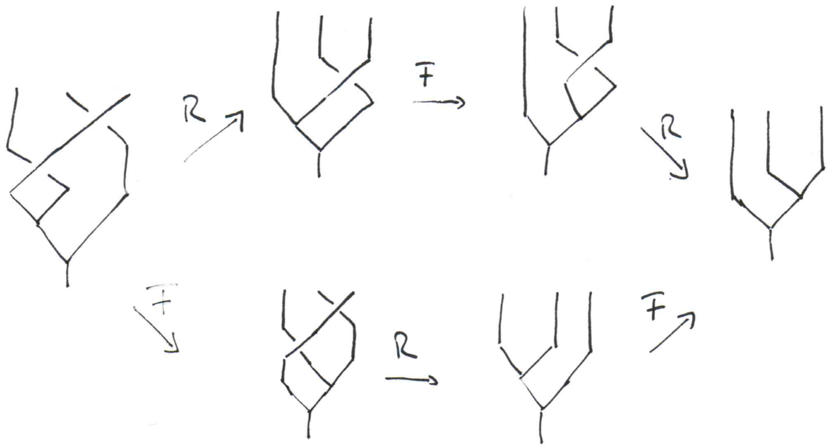
$$(F^k_{k_1,k_2,k_3})_{k_{23},k_{12}} = \left[\begin{array}{ccc} k_1 & k_2 & k_{12} \\ k_3 & k & k_{23} \end{array} \right].$$

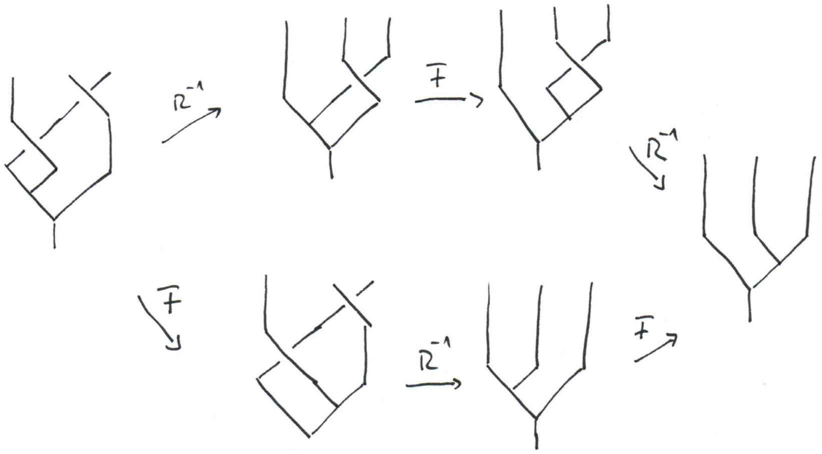
  Since we can have anyons without an underlying group, in general there may not be any Clebsch-Gordan coefficients but only an F*-matrix.*

9.4 The Pentagon and Hexagon equations

The following three equations – written in diagrammatic form and known as pentagon and hexagon equations – provide consistency conditions on the F and R-matrices.







MacLane's coherence theorem tells us that for an anyon model to be consistent it is sufficient that these three equations are satisfied.

9.5 Fibonacci anyons

The model we wish to construct has two distinct types of particles called 0 and 1 which obey the following fusion rules

$$0 \times 0 = 0 \qquad 0 \times 1 = 1 \qquad 1 \times 0 = 1 \qquad 1 \times 1 = 0 + 1.$$

Is there an F-matrix and an R-matrix compatible with these rules? Or are there even several different ones? In order to find out, we first solve the pentagon equation and find

$$F \equiv F^1_{111} = \begin{pmatrix} \tau & e^{i\varphi}\sqrt{\tau} \\ e^{-i\varphi}\sqrt{\tau} & -\tau \end{pmatrix},$$

where $\tau = \frac{\sqrt{5}-1}{2} = \phi - 1 \approx 0.618$ and ϕ is the golden ratio. $F^{\delta}_{\alpha\beta\gamma} = 1$ if at least one of the indices equals 0. To make life easier let us fix $\varphi = 0$. Solving the first hexagon equation then gives

$$R = \begin{pmatrix} e^{i\frac{4\pi}{5}} & 0 \\ 0 & e^{-i\frac{3\pi}{5}} \end{pmatrix}$$

and this turns also out to be consistent with the second one. Up to fixing the phase φ and exchanging the phases in the R matrix, this solution is unique. Let N^{α}_{n} be the number of paths when we fuse n anyons in the left standard basis and the final fusion label is α. Then

$$N^0_n = N^0_{n-2} + N^1_{n-2}$$

and since $N^1_{n-2} = N^0_{n-1}$ we find

$$N^0_n = N^0_{n-2} + N^0_{n-1}.$$

With the base cases $N^0_2 = 1$ and $N^0_3 = 1$ we find that $N^0_n = \text{Fib}(n-1)$, the $n-1$'th Fibonacci number. This property gives the model its name and it follows from a formula for the Fibonacci numbers that

$$N^0_n \approx \frac{\phi^{n-1}}{\sqrt{5}} = 2^{O(n)}.$$

The model therefore provides us with a large enough Hilbert space to do useful quantum computation.

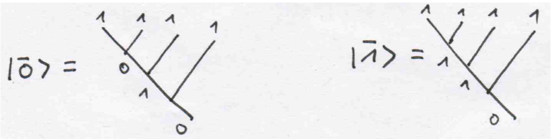
9.6 The Fibonacci anyons are universal for quantum computation

We have seen in Chapter 8 that the circuit model is universal for quantum computation. Rather than showing directly that the Fibonacci model is universal for quantum computation, we will show that within the Fibonacci model we are able to simulate the circuit model. The work splits into two parts:

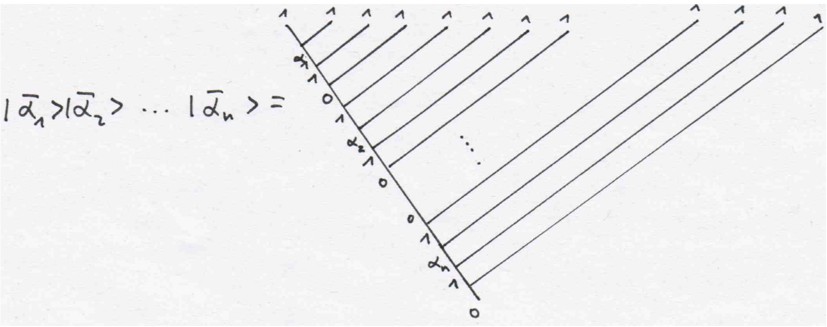
1. Encode qubits into fusion paths. The encoded qubits are known as logical qubits.

2. Construct braids that act as single qubit and CNOT operations on logical qubits.

1) We encode a single qubit into the two basis states of four anyons



By concatenating the trees n times, we can encode n qubits into $4n$ anyons:



2) When braiding the first and the second anyon the corresponding unitary transform on the first logical qubit is given by the R matrix

$$\tau_1 \mapsto R$$

In order to compute how braiding anyon two and three affects the logical qubit, we first need to carry out an F-move, then apply the R matrix and then invert the F-move. All in all

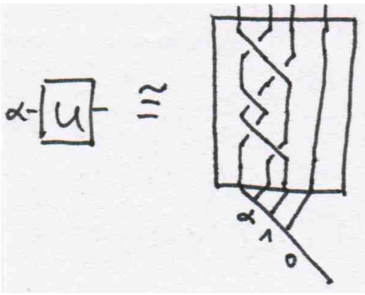
$$\tau_2 \mapsto B := FRF^{-1}.$$

This generalises immediately to the following action on the i'th logical qubit

$$\tau_{4i-3} \mapsto R$$

$$\tau_{4i-2} \mapsto B.$$

In the exercise we have seen how one can obtain any single qubit operation by a specific iteration of F and B.



Freedman, Larsen and Wang have generalised this observation and shown that by braiding n anyons one can approximate any unitary matrix (disregarding an overall phase factor) acting on the path labels, i.e. we can approximate all of $SU(N_n^0)$. When braiding 8 anyons we can therefore approximate any element in $SU(13)$, but in particular any element – especially the CNOT gate – of $SU(4)$ acting on the logical qubits $|\alpha_1\rangle\,|\alpha_2\rangle$.

This concludes the proof that the Fibonacci model can simulate the circuit model. Admittedly, such a simulation would be rather useless if an efficient circuit would be turned into an inefficient one by the simulation. So, we need to show that this is not the case. If we could simulate our single qubit and CNOT gates perfectly with only a finite number of gates, say c, then a circuit with m gates would be transformed into a braid with cm particle exchanges. Unfortunately, we can only approximate our gates with the braids which makes things a little more tricky.

Assume that we wish to simulate a circuit

$$U = U_m \cdots U_1$$

consisting of m gates by a braid

$$V = V_m \cdots V_1$$

that is obtained by replacing every gate U_i in the original circuit by its approximating braid V_i. Assume we are happy to tolerate a total error $\epsilon > 0$, i.e.

$$||U - V|| \leq \epsilon,$$

where $||X|| := \sup_{|\psi\rangle : \langle\psi|\psi\rangle = 1} |\langle\psi| X |\psi\rangle|$ is the operator norm of a matrix X. Then it follows from the triangle inequality that

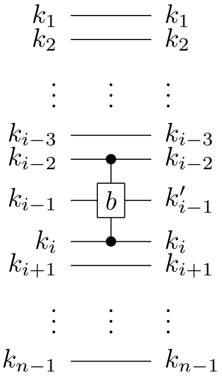
$$\begin{aligned}
||U - V|| &= ||U_m \cdots U_2 U_1 - V_m \cdots V_2 V_1|| \\
&= ||U_m \cdots U_2 U_1 - U_m \cdots U_2 V_1 + U_m \cdots U_2 V_1 - V_m \cdots V_2 V_1|| \\
&\leq ||U_m \cdots U_2 U_1 - U_m \cdots U_2 V_1|| + ||U_m \cdots U_2 V_1 - V_m \cdots V_2 V_1|| \\
&= ||U_1 - V_1|| + ||U_m \cdots U_2 - V_m \cdots V_2|| \\
&\vdots \\
&\leq \sum_{i=1}^{m} ||U_i - V_i||.
\end{aligned}$$

In general the use of the triangle inequality is tight and therefore, in order to get an error of ϵ for the entire circuit an error of at most ϵ/m is required for every individual gate. It can be shown that a gate can be approximated to error δ with a braid of length $O(1/\delta)$[15]. Setting $\delta = \epsilon/m$ we see that we can simulate a circuit consisting of m gates with a total error of ϵ with a braid of total length $O(m^2/\epsilon)$. A circuit that is of polynomial size in the input length – that is efficient – will therefore be transformed into a braid that is also of polynomial length in the input. A fundamental result in quantum computation, the Solovay-Kitaev theorem, states that this result can even be improved and that only a circuit of size $O(m \log^c \frac{m}{\epsilon})$ is needed, where c is some constant between one and two.

[15] For a related argument that shows that single qubit gates can be approximated by the $\pi/8$ gate and the Hadamard gate see Nielsen and Chuang

9.7 Simulating the Fibonacci model within the circuit model

Just as we were able to simulate the permutational quantum computer in the circuit model, we can simulate the Fibonacci model within the circuit model. This is easily seen by noting that the action of the braid group on the path labels takes the following form: The generator τ_i is mapped to



where the coefficients of the matrix b are easily calculated from the F and R matrices. A braid can thus be seen as a circuit consisting of three-qubit gates which can be expressed *exactly* as single qubit and CNOT gates. This argument can readily be generalised to any other anyon model.

9.8 Truncating $SU(2)$

Unfortunately, the most natural algorithm for the topological quantum computer is not formulated for the simple Fibonacci anyon model, but for the following, more complicated, models.

These are truncations of the $SU(2)$ ”anyon“ model we have studied. For each truncation parameter $k \in \mathbb{N}$, the anyon model called $SU(2)_k$ has particle labels $\{0, 1, \ldots, k\}$ and fusion rules just as in $SU(2)$:

$$r_1 \times r_2 = \sum_{r=|r_1-r_2|:r \mod 2 = r_1+r_2 \mod 2}^{\min(r_1+r_2, 2k-r_1-r_2)} r$$

Example 3. *$SU(2)_2$ has fusion rules*

$$1 \times 1 = 0 + 2$$

$$1 \times 2 = 1$$

$$2 \times 2 = 0.$$

All other rules follow from 0 being the trivial label.

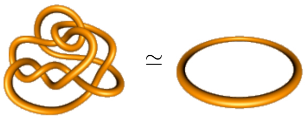
The F- and R-matrix are not unique given these fusion rules, but there is a solution that can be obtained by *deforming* the $SU(2)$ Clebsch-Gordan coefficients (see J. Slingerland's PhD thesis, University of Amsterdam). The resulting model is known as $SU(2)_k$ (in words: $SU(2)$ at level k).

9.9 Knots, Links and Braids

A knot is a closed non-intersecting curve in $\mathbb{R}^3$.

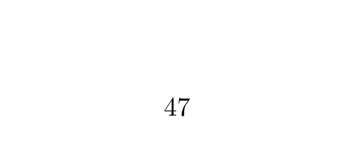
Instead of working with knots we choose to work with links, that is with curves in $\mathbb{R}^3$ consisting of several knots.

A link may be oriented by assigning a direction to each of its components. Two links are equivalent (isotopic) if they can be deformed into each other.

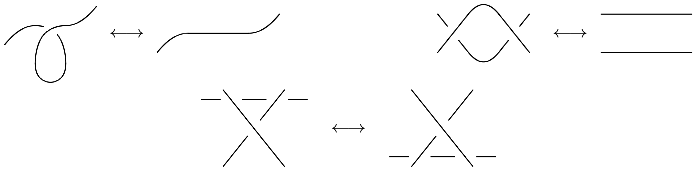


Problem: Given two (oriented) links, are they equivalent?

In order to have a better handle on links, we will represent links by projecting them to $\mathbb{R}^2$ but recording which of two strands goes above the other in a crossing.



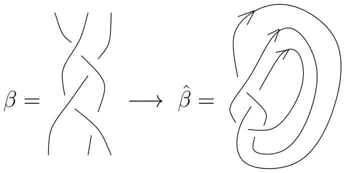
It turns out that two links are equivalent if and only if their representations are related by Reidemeister moves.



Unfortunately, applying Reidemeister moves to test equivalence is a computationally costly business, as there is no polynomial upper bound on the number of additional crossings known.[16]

A different way to test equivalence is to associate an invariant to each link, that is an algebraic object (e.g. a number, a polynomial) that stays invariant under smooth deformations of the link. If two links have different objects associated to them, then they cannot be equivalent. Most invariants are, however, not complete, meaning that there are inequivalent links that have identical objects associated to them.

It is our goal to associate an invariant number to each oriented link. We will do this by associating a specific number to a braid. So, we first have to establish a connection between links and braids:[17]



We say that two braids β and β' are equivalent if $\beta' = \alpha\beta\alpha^{-1}$ or $\beta' = \beta\tau_n^{\pm 1}$ (in B_{n+1}).

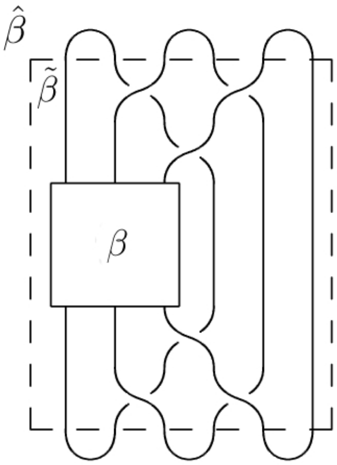
Theorem 5 (Alexander)**.** *Every oriented link can be written as a closure of a braid.*

Theorem 6 (Markov)**.** *Two oriented links are equivalent if and only if the corresponding braids are equivalent.*

We write the closure operation in the following way:

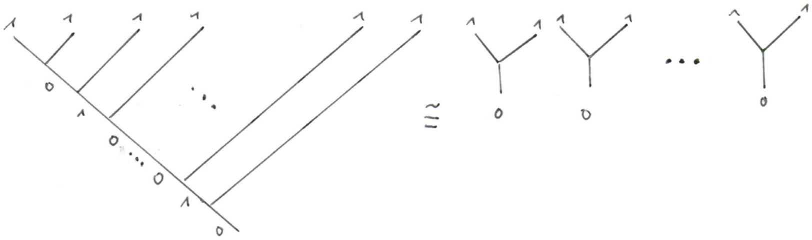
[16] Note that one of the moves corresponds to the Yang-Baxter equation.

[17] The following graph is taken from [5].



9.10 A quantum algorithm for approximating the Jones polynomial

We interpret the top part as the input state $|\psi\rangle$:



Denoting the entire braid by $\tilde{\beta}$ and the corresponding unitary by $U_{\tilde{\beta}}$, we associate the probability of obtaining outcome $|\psi\rangle$,

$$p := |\langle\psi| U_{\tilde{\beta}} |\psi\rangle|^2,$$

to the link. By Markov's theorem (and the fact that we have a representation of the braid group), this number is an invariant under smooth deformations of the link. When working in the anyon model $SU(2)_k$, this number is usually denoted by

$$p = \frac{|J(\hat{\beta}, e^{\frac{i2\pi}{k+2}})|^2}{|J(n \text{ unknots}, e^{\frac{i2\pi}{k+2}})|^2}$$

where $J(\hat{\beta}, q)$ ($q \in \mathbb{C}$) is the *Jones polynomial*, a famous invariant of oriented links. The right hand side is between 0 and 1, because $|J(\hat{\beta}, e^{\frac{i2\pi}{k+2}})| \leq |J(n \text{ unknots}, e^{\frac{i2\pi}{k+2}})|$.

By running our anyon computer n times with this input and braid, we obtain a sequence of independently and identically distributed random variables. Each random variable takes value 1 with probability p and value 0 with probability $1-p$. By the law of large numbers (here in its incarnation by Hoeffding), for all $\delta > 0$,

$$\mathrm{Prob}\left(|X^n - p| > \delta\right) = \mathrm{Prob}\left(|X^n - \frac{|J(\hat{\beta}, e^{\frac{i2\pi}{k+2}})|^2}{|J(n \text{ unknots}, e^{\frac{i2\pi}{k+2}})|^2}| > \delta\right) \leq e^{-2n\delta^2},$$

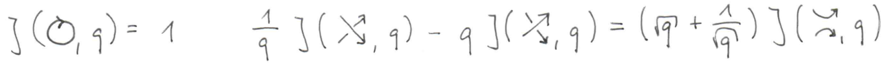
where $X^n := \frac{\sum_{i=1}^n X_i}{n}$ is the observed average outcome of the measurement.

In order to approximate the normalised Jones polynomial to accuracy $\delta > 0$ (with arbitrarily high probability) one therefore needs a number of operations that is efficient (i.e. polynomial) in the number of crossings with which the braid is represented. But how good an approximation of $J(\hat{\beta}, e^{\frac{i2\pi}{k+2}})$ is it? In order to find out we first need a proper definition of the Jones polynomial.

9.11 Jones polynomial

We have seen that the values of the Jones polynomial at roots of unity appeared very naturally in our setup. In fact, Jones came to discover his polynomial in a way very similar to this. Instead of certain anyon models he used the concept of a Hecke algebra (which depends on a parameter $q \in \mathbb{C}$) on which he represented the braid group.

Researchers have also used an inductive definition of the Jones polynomial via *skein relations*:



where the arguments in the equation on the right stand for links that are identical except one crossing, which is shown as argument. This way of defining the Jones polynomial has a two-variable generalisation, the HOMFLYPT polynomial. Both ways of defining the Jones polynomial are discussed in [5].

For a calculation of the normalisation constant we use the second definition of the Jones polynomial and find

$$|J(n \text{ unknots}, e^{\frac{i2\pi}{k+2}})| = \left(2\cos\frac{\pi}{k+2}\right)^{n-1}.$$

which is exponentially increasing in n for $k \geq 2$.[18] Hence, our algorithm does not give a good approximation of the Jones polynomial itself, but only its normalised version. This may be slightly disappointing, but we should still keep in mind that the algorithm we constructed is exponentially faster than any known classical algorithm for this problem.

[18] The website `http://library.thinkquest.org/12295/data/Invariants/Articles/Jones.html` computes the Jones polynomials of a few links explicitly.

9.12 Complexity theory

In order to put our algorithm for approximating the Jones polynomial into perspective, we will take a detour and say a few words about complexity theory.

In complexity theory, functions whose input is a bit string that depends on a length n are grouped into complexity classes according their computational difficulty, viz. according to the minimal number of gates required in a circuit for f, asymptotically.

Informally, the two most prominent complexity classes can be defined as follows.

- The complexity class P consists of all problems that can be computed with circuit of polynomial size in the length of the input.
- The complexity class NP (Nondeterministic Polynomial-Time) consists of the problems such that a possible answer to an input of length n can be verified with a circuit of polynomial size.

Note that $P \subseteq NP$. It is the biggest open problem in theoretical computer science, whether or not $P = NP$, but it is strongly conjectured that $P \neq NP$. Examples of problems in P are

- Multiplying: multiply two integers of combined bit length n (of which degree is the polynomial $g(n)$ which is implied by the algorithm that you learned in primary school?)
- Primes: determining whether or not an n-bit number is prime

Problems in P are called *efficient*. Note that this is the definition of the word efficient in computer science. Intuitively, we may only want to think of problem as efficient if it is $O(n)$, $O(n^2)$ or maybe $O(n^3)$. Interestingly, most relevant problems in P have a low exponent and thus our intuition of efficient sort of coincides with the definition of P. Note also that someone performing actual calculations on a computer, does not have arbitrary resources and hence he may consider a program with runtime 0.00001×2^n as being more efficient as one with $10^{10} \times n^2$. The constant that we wipe under the carpet in our theoretical analysis may therefore be very relevant in practice.

Problems in NP that are not known to be in P are

- Factoring: factor an n-bit number into its primes (this problem is clearly in NP, since multiplying is in P)
- Traveling Salesman: determine the shortest route between a number of cities such that every city is visited exactly once.
- SAT: given a Boolean formula, i.e. a formula containing $\vee, \wedge, \bar{\ }$, brackets and variables, is there an assignment of the Boolean variables x_i such that the formula is true?

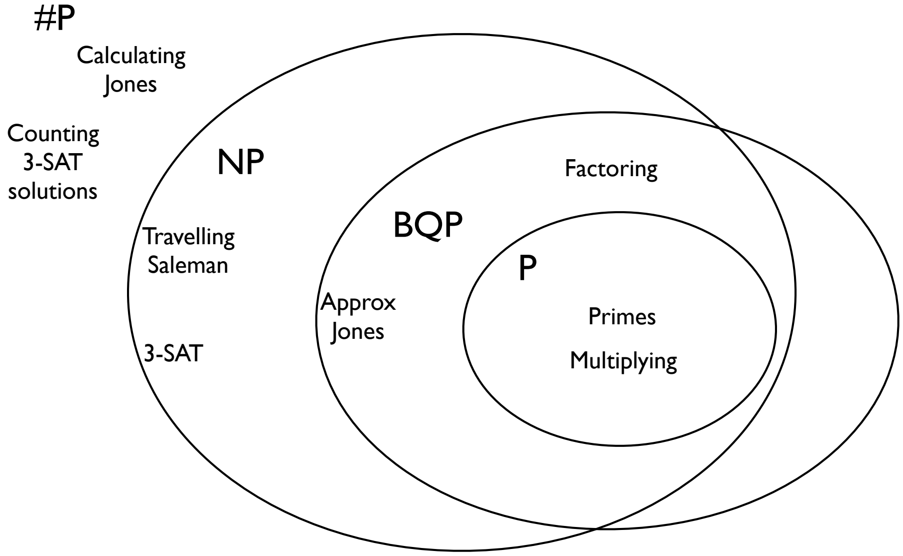


Figure 4: Problems in the complexity classes P, BQP and NP. A problem shown on the border of a complexity class is *complete* for this class.

A problem is NP-hard, if, given a black box (or oracle) that solves the problem, we can solve any other problem in NP in polynomial time. A problem is NP-complete if it is *in* NP and NP-hard. It is the famous Cook-Levin theorem that asserts that SAT is NP-complete. NP-complete problems are hence the most difficult problems in NP. For an illustration see Figure 4.

The most important result in quantum complexity theory is Shor's factoring algorithm, which shows that factoring is efficient on a quantum computer, whereas we do not know of an efficient classical algorithm for the problem. The best known classical algorithm (general number field sieve algorithm) runs in time $O\left(\exp\left((\frac{64}{9}n)^{\frac{1}{3}}(\log n)^{\frac{2}{3}}\right)\right)$. Factoring is thus in NP and in BQP (bounded-error quantum polynomial time), the quantum analog to P. There are also many other complexity classes; for a good overview see `http://qwiki.stanford.edu/index.php/Complexity_Zoo`. Let me highlight one more class, known as #P (pronounced "sharp"-P). Whereas NP consists of all problems for which a given solution is easily verified, #P consists of all problems which can be represented as counting the number of solutions to a problem in NP. Counting problems are considered to be much more difficult than problems in P or NP although formally it is an open problem to prove that #P is different from NP or P.

In this course, we have seen that approximating the Jones polynomial is in BQP, that is, we have exhibited a quantum algorithm that can approximate the Jones polynomial in time polynomial in the number of crossings of the link. This is already exciting as we do not know of a classical algorithm for this problem. More so, the problem turns out to be BQP-complete, that is, the most difficult problem in BQP. Formally, this means that given access to an oracle that approximates the Jones polynomial, we can solve any other problem in BQP in polynomial time on a classical computer. Let us see how this works:

A problem in BQP has by definition a polynomially-sized circuit that solves it. Encode this quantum circuit into a braid. Then the probability for the circuit returning the answer yes[19] can be expressed as the (normalised) Jones polynomial of the braid. Hence, if we have an oracle that approximates the Jones polynomial, we can approximate the probability of the answer yes of the quantum circuit and hence solve the problem.

This has an interesting consequence for factoring integers (or any other problem in BQP for that matter): If we had an efficient classical algorithm that approximates the Jones polynomial, we could regard this algorithm as our oracle and use it to convert Shor's efficient quantum algorithm for factoring into an efficient classical algorithm.

As shown in the figure, calculating the Jones polynomial rather than its normalised version is #P-complete. Since #P is not believed to be equal to P (nor BQP), calculating the Jones polynomial can be regarded as computationally intractable.

10 Acknowledgements

I thank Mario Berta for help in setting up this course and Stephen Jordan and Jon Yard for letting me use their figures in these notes. My understanding of the material presented in the first part of the course was shaped by discussions with Graeme Mitchison and in the second part by discussions with Jon Yard.

References

[1] D. Bacon, I. Chuang, and A. Harrow, *Efficient Quantum Circuits for Schur and Clebsch-Gordan Transforms*, Physical Review Letters, 97 (2006), pp. 1–4.

[2] E. Battey-Pratt and T. Racey, *Geometric Model for Fundamental Physics*, Int. J. Th. Ph., 19 (1980), p. 437.

[3] C. Bennett, G. Brassard, C. Crepeau, R. Jozsa, A. Peres, and W. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Physical Review Letters, 70 (1993), pp. 1895–1899.

[4] A. W. Harrow and H.-k. Lo, *A Tight Lower Bound on the Classical Communication Cost of Entanglement Dilution*, IEEE Transactions on Information Theory, 50 (2004), pp. 319–327.

[19] we are only concerned with decision problems which have either yes or no as answer

[5] V. F. R. Jones, *The Jones polynomial*, (2005). manuscript.

[6] S. P. Jordan, *Permutational Quantum Computing*, Quantum Information and Computation, 10 (2010), pp. 0470–0497.

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