QUANTUM COMPUTATION

FERNANDO LIU LOPEZ

Contents

(Work in progress) These are my notes on Quantum Computation. They are based primarily on Ladsberg's *Quantum Computation and Quantum Information*, along with Heunen and Vicary's notes on *Categorical Quantum Mechanics* for insights into underlying Quantum Mechanics discussed through a mathematical lens.

Prerequisites: Knowledge of Linear Algebra is a must to read this. Some familiarity with Hilbert spaces and/or Functional Analysis is useful, though most of that terminology will be reintroduced here. Familiarity with abstract algebra (groups, rings, fields, modules), basic coding, category theory and monoidal categories is useful to understand *my* perspective, but can be skipped for the most part.

1. Primer on Classical Computation

1.1. Introduction.

Definition 1.1. A **bit** (binary digit) is a unit of information that can take on two possible values (or **states**), namely 0 or 1. We denote bits by $x \in \mathbb{F}_2$. A **bit-string** of length n is a tuple $x = x_0x_1 \cdots x_{n-1} \in \mathbb{F}_2^n$ (here each $x_i \in \mathbb{F}_2$). A bit-string length n has 2^n possible states, since $|\mathbb{F}_2^n| = \Pi_{i=1}^n |\mathbb{F}_2| = 2^n$.

Remark 1.2 (Implementing bits). Bits can be implemented through any physical process which we can split into two possible states: positions of a light switch, two different voltage levels, two distinct levels of light intensity, two different directions of polarization, etc.

Remark 1.3 (What is classical computation?). The **theory of computation** is the branch of computer science that deals with problems that can be solved using **algorithms**: a finite set of unambiguous instructions and end conditions that, given some initial conditions, can be performed in a prescribed sequence to achieve a certain goal.

The bit is the basic unit of information for classical computation. As such, classical computation translates input/ouput data into bit-strings, and translates pairs:

$$(\text{initial conditions, desired solution}) \rightsquigarrow (x \in \mathbb{F}_2^n, y \in \mathbb{F}_2^m).$$

Thus, problems are modelled by **Boolean functions** $\mathbb{F}_2^n \to \mathbb{F}_2^m$. Solving a problem amounts to creating a machine that implements the evaluation of such a function.

The following questions naturally arise:

- (**Q0**) Computability: Which problems can be solved?
- (**Q1**) Universality: Can we decompose $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ into simpler building blocks? In fact, can we find *universal* building blocks that allow us to build any $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$?
- (**Q2**) Complexity: If we attach costs to the different building blocks, steps, or subroutines (e.g. time, memory, and/or operational costs), can we build machines that implement solutions at a certain cost?

(**Q1**) will be addressed at several points. We largely ignore (**Q0**) and focus on (**Q2**) only when comparing quantum algorithms to classical counterparts.

Example 1.4 (Complexity for list search). Searching for an element in a list of size n by checking each index incrementally (**linear search**) gives a worst case runtime of $\mathcal{O}(n)$. Quantum algorithms like Grover's algorithm can find the elements in unsorted lists in $\mathcal{O}(\sqrt{n})$.

Example 1.5 (Time complexity and input size). It takes $1 + \log(n)$ bits to encode a positive integer n in binary (you can add an extra bit to record $\pm$ signs too). When computing complexity, it's important to remember that an input of n will have an input size of $\log(n)$. An algorithm that runs in polynomial time relative to the size of the input will thus have to be a polynomial in $\log(n)$.

1.2. **Classical Gates.** This section addresses the question of universality (**Q1**).

Definition 1.6. A **(Boolean) circuit** is a pictoral decomposition of a Boolean function. Formally, it is a finite acyclic directed graph with:

- a collection of vertices of in-degree zero called **inputs**,
- a collection of vertices of out-degree zero called **outputs**,
- and a collection of vertices in between of positive in/out degree called **gates**.

The graph is finite because machines we build have physical limits on the size of input/subroutines/outputs they can be built to handle. The graph is acyclic and directed to clearly establish the order/sequence in which operations are performed. Circuits are usually drawn with arrows pointing left to right. The gates in a circuit are the (simpler) building blocks (subroutines) through which we decompose our circuits.

Definition 1.7. A **gate set** is a collection of gates. A gate set is **universal** if any Boolean function can be computed with a circuit whose gates are labeled by the gate set.

Definition 1.8 (Elementary Classical Gates). The following gates are designated to have unit computational cost:

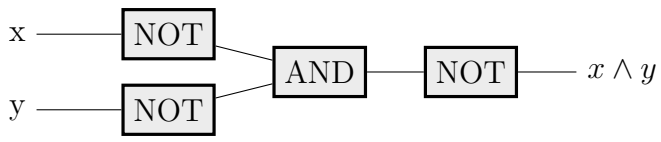
$$\begin{aligned}
\text{NOT}(\neg) &: \mathbb{F}_2 \to \mathbb{F}_2 & \neg x &= x + 1 \mod 2. \\
\text{XOR}(\oplus) &: \mathbb{F}_2^2 \to \mathbb{F}_2 & x \oplus y &= x + y \mod 2. \\
\text{AND}(\wedge) &: \mathbb{F}_2^2 \to \mathbb{F}_2 & x \wedge y &= xy \mod 2. \\
\text{OR}(\vee) &: \mathbb{F}_2^2 \to \mathbb{F}_2 & x \vee y &= x + y - xy \mod 2.
\end{aligned}$$

These gates contain redundancies. DeMorgan's Laws connect OR and AND using NOT:

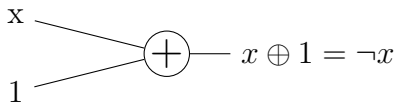
$$x \vee y = \neg(\neg x \wedge \neg y) \quad \text{and} \quad x \wedge y = \neg(\neg x \vee \neg y),$$

XOR can be expressed as $x \oplus y = (x \wedge \neg y) \vee (\neg x \wedge y)$, and NOT can be viewed as XOR-ing with 1 in the second slot.

Example 1.9. Here is a circuit implementing OR using DeMogan's Laws.



Example 1.10. We can also implement functions $\mathbb{F}_2^n \to \mathbb{F}_2^m$ using circuits that add more bits to the domain or codomain. We call these **workplace bits**. In some cases, these bits also begin at predetermined values. Here is NOT implemented with XOR and an extra workplace bit $\mathbb{F}_2^{1+1} \to \mathbb{F}_2$.



Question 1.11. The formula $x \vee y = x + y - xy$, shows that we should be able to implement OR using XOR and AND. Can you draw such a circuit using only XOR and AND?

Remark 1.12 (COPY and DELETE). In greater detail, OR decomposes as:

$$(x, y) \xrightarrow{\Delta \times \Delta} (x, x, y, y) \xrightarrow{1 \times \sigma \times 1} (x, y, x, y) \xrightarrow{\oplus \times \wedge} (x + y, xy) \xrightarrow{\oplus} x + y - xy.$$

The diagonal map $\Delta : \mathbb{F}_2 \to \mathbb{F}_2 \times \mathbb{F}_2$ is the COPY operation $x \mapsto (x, x)$, while σ is the FLIP operation sending $(x, y) \mapsto (y, x)$. Categorically, classical computation can be thought of as living in $(\mathbf{Set}, \times, \bullet)$ (or $\mathbf{FSet}$): the Cartesian category of sets, where:

(1) the monoidal product is given by the categorical product (Cartesian product),

(2) the monoidal unit is a singleton set denoted $\bullet$.

It is a feature of classical computation that we always have canonical COPY and DELETE maps (DELETE is given by terminal maps $\epsilon : X \xrightarrow{!} \bullet$). These maps are canonical in the sense that sets have unique (cocommutative) comonoidal structures, with coproduct and counit are given by the COPY and DELETE maps (in particular, the counit laws force Δ to be the diagonal map). Quantum computation will be "fundamentally" different than classical computation, in part due to the fact that the categories of quantum computation won't have canonical COPY and DELETE.

The unique comonoidal structures of sets is also why generalizing the structure of groups to Hopf algebras might initially feel unnatural (groups are Hopf algebras in $\mathbf{Set}$). Unsweeping the comonoidal structure from under the rug gives:

Group: set + (product + unit) + $\exists!$**(coproduct + counit)** + inversion,

Hopf Algebra: object + (product + unit)+(coproduct + counit) + antipode/inversion,

which makes the generalization much more obvious.

With regards to the FLIP map, the FLIP in **Set** satisfies $\sigma^2 = 1$, making **Set** a symmetric monoidal category. However in certain quantum mechanical applications, physical processes will have nontrivial flips, e.g. when two fermions are swapped their wave equations gain a phase factor of -1.

Proposition 1.13. The gate set {NOT, AND, OR, COPY} is universal.

Proof. Start with $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ sending $x \mapsto (f_1(x), \ldots, f_m(x))$. By copying the input m times, we can focus on building the component functions instead. Thus WLOG assume $f : \mathbb{F}_2^n \to \mathbb{F}_2$.

Let $a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$ be an input such that $f(a) = 1$. We will build indicator functions identifying all such a's. Since XOR can be constructed from this gate set, define $\phi_{a_i}(x) = x \oplus a_i$. Next, define $\phi_a : \mathbb{F}_2^n \to \mathbb{F}_2$ by:

$$\phi_a(x_1, \ldots, x_n) = \bigwedge_{i=1}^{n} \phi_{a_i}(x_i) = \phi_{a_1}(x_1) \wedge \cdots \wedge \phi_{a_n}(x_n).$$

Notice that $\phi(x) = 1$ iff $x = a$. Finally, notice that:

$$f(x) = \bigvee_{a \in \mathbb{F}_2^n : f(a) = 1} \phi_a(x),$$

since the right hand side returns 1 on input x iff $\exists a \in \mathbb{F}_2^n$ with $f(a) = 1$ for which $x = a$. □

1.3. **Reversibility.** Reversible classical computations deals only with invertible Boolean functions $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$. We explore them because quantum circuits will be required to be reversible to be in line with physical laws of quantum mechanics. However, even in classical computation, reversible gates are preferred, because bit-erasure is always accompanied by heat dissipation, which is not good for the machines implementing the circuits. The cost of gaining reversibility will be the need for more workplace bits.

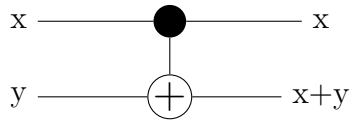
Example 1.14. NOT is reversible and involutive since $\neg\neg x = x$.

Example 1.15 (Our universal gate is terrible). Of our universal gate set:

$$\{\text{NOT, AND, OR, COPY}\},$$

the AND and OR operations are not reversible. Reversing the COPY operation would still result in an erased bit, which defeats the purpose of reversibility.

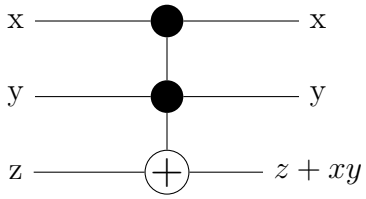
Example 1.16. The **controlled not** gate CNOT is reversible and involutive $\mathbb{F}_2^2 \to \mathbb{F}_2^2$. It uses the first input to decide whether or not to negate the second input, sending $(0, y) \mapsto (0, y)$ and $(1, y) \mapsto (1, \neg y)$. It is denoted by:



Example 1.17. The **Toffoli gate** $\text{Tof} : \mathbb{F}_2^3 \to \mathbb{F}_2^3$ with:

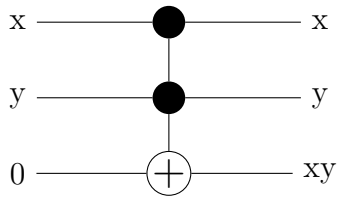
$$(x, y, z) \mapsto \big(x, y, z \oplus xy\big)$$

is also reversible and involutive: $\text{Tof} \circ \text{Tof} = id$. Tof can be understood as a 3-bit analog to CNOT, where one of either x or y can be 1 to trigger the third slot to switch.

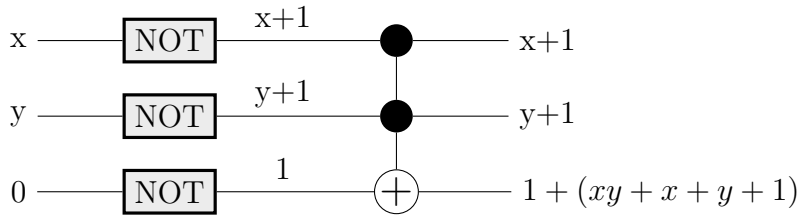


Proposition 1.18. The gate sets $\{\text{Tof}, \text{CNOT}, \neg\}$ and $\{\text{Tof}, \neg\}$ are universal and reversible. In particular, reversible classical computation allows us to solve the same problems as classical computation.

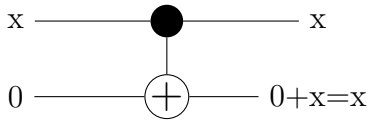
Proof. All we really need to show is how to implement AND, OR, and COPY using the Toffoli gate. AND can be implemented with three workplace bits as $\mathbb{F}_2^{2+1} \to \mathbb{F}_2^{1+2}$ by setting $z = 0$, giving $(x, y, 0) \mapsto (x, y, xy)$.



To get OR, simply add NOT gates to all inputs at the start:



In particular, the bottom register simplifies to $xy + x + y = x + y - xy = x \vee y$. Finally, COPY can be obtained from CNOT by:



and thus can be obtained from Tof as well. □

1.4. **Linear Algebra.** We can encode classical computation via linear algebra by replacing $\mathbb{F}_2 = \{0, 1\}$ with $\mathbb{R}^2$ equipped with some choice of basis, which we denote by $|0\rangle, |1\rangle$. When working with multiple bits, we will often use the isomorphism $(\mathbb{R}^2)^{\otimes n} = \mathbb{R}^2 \otimes \cdots \otimes \mathbb{R}^2 \cong \mathbb{R}^{2^n}$. For $(\mathbb{R}^2)^{\otimes n}$ with the $|0\rangle, |1\rangle \in \mathbb{R}^2$ basis, we get an induced basis $|I\rangle := |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle$ with $i_\alpha \in \{0, 1\}$. For example:

$$\text{A 2-bit is a basis element } |00\rangle, |01\rangle, |10\rangle, |11\rangle \text{ of } \mathbb{R}^2 \otimes \mathbb{R}^2.$$

For the $\mathbb{R}^{2^n}$ representation write bases as $|i\rangle$ for $1 \leq i \leq 2^n$ or $0 \leq i \leq 2^n - 1$. For example:

$$\text{A 2-bit is a basis element } |0\rangle, |1\rangle, |2\rangle, |3\rangle \text{ of } \mathbb{R}^4.$$

Both representations use lexicographic orderings on their basis elements to encode coordinate vectors and matrix representations.

Remark 1.19 (Permutation matrices encode classical computation). Since classical computation deterministically sends some n-bit to some m-bit, the corresponding matrix representations will always be given by **permutation matrices**. These are invertible and hence reversible as well.

Remark 1.20 (Implementation constraints). Since $\{\text{Tof}, \neg\}$ is a universal reversible gate set, we will focus on gates/matrices that, at most, act non-trivially on some copy of $(\mathbb{R}^2)^{\otimes 3}$ (the Toffoli gate acts on three bits). Think of this constraint as a small favor we're

doing to the people building the physical circuits. This constraint will become "necessary" in quantum computing, since instantiating quantum gates is currently pretty hard.

Example 1.21 (Matrix of NOT). The NOT gate on a single bit corresponds to:

$$\text{NOT} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

because it takes $|0\rangle \mapsto |1\rangle$ and $|1\rangle \mapsto |0\rangle$.

Example 1.22 (Matrix of CNOT). Work with two bits using the $\mathbb{R}^2 \otimes \mathbb{R}^2$ representation. The basis vector $|01\rangle$ comes third in the lex order, so it is represented by the column vector $(0, 0, 1, 0)^t$.

The CNOT gate has 2-bits as input and output, so it's represented by a map $\mathbb{R}^{2^2} \to \mathbb{R}^{2^2}$. Its action on the basis vectors is given by:

$$|00\rangle \mapsto |00\rangle, \quad |01\rangle \mapsto |01\rangle, \quad |10\rangle \mapsto |11\rangle, \quad |11\rangle \mapsto |10\rangle,$$

so it's represented by the matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Interlude on Quantum Mechanics

This is a short interlude giving a rough sense of the definitions and results that will guide our shift from classical to quantum computation.

In classical mechanics, a **state** consists of a list of variables with fixed real values at any point in time. The values of the state evolve deterministically over time under the laws/equations of motion. A **measurement** consists of some physical process that extracts/reads off this data at some point in time. An **observable** a physical quantity that can be measured, and thus can be thought off as a real-valued function on the set of all possible states.

In quantum mechanics, **states** similarly consist of list of variables, whose values are/can be assumed to be complex, since experimental results (e.g. double slit experiment) showcase behaviors such as quantum interference which can be explained mathematically by incorporating phases $e^{i\theta}$. One representation of quantum states is via **wave functions** ψ. The **superposition principle** states that these can be added, scaled, or even integrated, which has led to a more abstract considering of quantum states as elements $|\psi\rangle$ of separable Hilbert spaces (e.g. think wave functions as elements of L^2).

Unitarity is a postulate of quantum mechanics, stating that time evolution of a quantum state is mathematically represented by unitary operators. **Stone's theorem** establishes a correspondence between one parameter families of strongly-continuous unitary operators $(U_t)_{t\in\mathbb{R}}$ and self-adjoint (Hermitian) operators A via $U_t = e^{itA}$. **Schrodinger's equation**:

$$i\hbar\frac{d|\psi\rangle}{dt} = A|\psi\rangle$$

explicitly gives the time evolution of a wave function in terms of a fixed self-adjoint operator A called the **Hamiltonian** of the system.

Informally, **Born's rule** states that the probability of an outcome is the square of the amplitude. In more detail, it identifies an **observable** of a system in (normalized) state $|\psi\rangle$ as a self-adjoint operator A. In the case when the spectrum of A is discrete, the measured result will be an eigenvalue λ_i of A (which are always real), with a certain probability. The probability of observing λ_i is $\langle\psi\,|P_i|\,\psi\rangle$, where P_i is the projection onto the eigenspace of λ_i. In the case when the eigenspace is one-dimensional and spaced by the normalized eigenvector

$|\lambda_i\rangle$, the projection is given by $|\lambda_i\rangle\langle\lambda_i|$, so the probability is $\langle\psi\,|\,\lambda_i\rangle\langle\lambda_i\,|\,\psi\rangle = |\langle\lambda_i\,|\,\psi\rangle|^2$. Measurement filters/collapses the quantum system, so that subsequent measurements of an isolated system will continue to give identical values.

The **complementarity** principle states that some pairs of observables cannot be measured simultaneously. Mathematically this is encoded by the non-commutativity of their corresponding operators. Observables are **compatible** if they commute. Heisenberg's **uncertainty principle** states that position and momentum are incompatible. A desirable feature is thus having a complete set of compatible observables, so that we may "prepare" quantum systems to be in specified states by measuring them until we reach a desired initial state.

2. Towards Quantum Computation

The rough idea of quantum computation is using quantum states as states for a new kind of computer. Since quantum states change according to the laws of quantum mechanics, we will use quantum mechanical principles and postulates to guide the construction of our theory of quantum computation.;

2.1. **Probabilistic Computation.** Classical (deterministic) computations took bit-strings $x \in \mathbb{F}_2^n$ to other bit-strings $f(x) \in \mathbb{F}_2^m$. In terms of linear algebra, bits were encoded as basis vectors $|I\rangle \in \mathbb{R}^{2^n}$ and operators were represented by permutation matrices.

However, one feature of quantum states (superposition principle) was that states could be added and scaled, so they could be understood as elements of vector spaces. Our linear algebraic model of classical computation allowed for an interpretation of states as *basis vectors* in a vector space, but didn't have a physical interpretation for linear combinations of these states.

Furthermore, Born's rule states that observable outcomes of measuring a quantum state are probabilistic, not deterministic. Specifically, given an eigenbasis for an observable, the (squared) magnitudes of a state's coefficients wrt the eigenbasis defined a probability distribution determining the probability of observing certain values.

This suggests switching from Boolean functions to functions that map an input bit string to a probability distribution on the possible output bit-strings. This is what we'll call **probabilistic computation**.

For example, a classical coin flip can thought of as a Boolean function $\mathbb{F}_2 \to \mathbb{F}_2$ mapping the visible face of the coin pre-flip to the one post-flip. How this function is defined will only be determined once the coin is actually flipped. However, probabilistically, we know that a fair coin flip takes the current state (the side of the coin currently facing up) to any of the two faces, with equal probability. This amounts to associating to both heads and tails the uniform distribution on $\{\text{heads}, \text{tails}\}$.

Probabilistic computations can be modeled as functions $\mathbb{F}_2^n \to \mathbb{R}^{2^m}$. Picking a basis $|I\rangle$ for the range $\mathbb{R}^{2^m}$, a probabilistic computation maps a bit-string $I_0 \in \mathbb{F}_2^n$ to a **convex combination** $\Sigma_I p_I |I\rangle$. Here the coefficients satisfy $p_I \in [0, 1]$ and $\Sigma_I p_I = 1$, so they can

interpreted as probability distributions on $\mathbb{F}_2^m$ via $P(X = I) = p_I$ (the probability that our random variable takes on the value I is p_I).

We may also pick a basis $|I\rangle$ for $\mathbb{R}^{2^n}$ and embed $\mathbb{F}_2^n \to \mathbb{R}^{2^n}$ via $I \mapsto |I\rangle$. This allows us to model probabilistic computations as functions $\mathbb{R}^{2^n} \to \mathbb{R}^{2^m}$, which we can represent using linear algebra. The matrices of probabilistic computations will be **stochastic**: column entries will be non-negative and will sum to 1 (Markov chains). Each $|I_0\rangle \mapsto \Sigma_I p_I |I\rangle$, where p_I is the probability that an input of I_0 will result in an output of I.

Finally, notice that inputs of form $v = \Sigma_I p_I |I\rangle$ also have physical interpretations, as probability distribution vectors of previous probabilistic computations whose results we don't have access to directly.

Example 2.1. Given any basis $|0\rangle, |1\rangle$ of $\mathbb{R}^2$, a coin flip is represented by the matrix:

$$\text{COIN} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad |0\rangle \mapsto \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle \quad \text{and} \quad |1\rangle \mapsto \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle.$$

Notice that this computation is not reversible: $\det(\text{COIN}) = 0$. Although the physical state of the coin will always be one $(1, 0)^t$ or $(0, 1)^t$ in coordinates, we can interpret an input of $(1/2, 1/2)^t$ as modeling a coin that has been flipped and handed to us while we keep our eyes closed. Flipping the coin again with our eyes closed is modeled by:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}.$$

Remark 2.2 (Categorical perspective on probabilistic computation). Probabilistic computation takes place in the Kleisli category of the distribution monad.

Remark 2.3 (Approximating Probabilities). In classical probabilistic circuits over $\mathbb{F}_2^n$, we don't have true randomness. Instead, independent random coin flips are used to mimic randomness. If we use r bits to model randomness, there are 2^r possible states with uniform probability of being chosen. We can approximate other probability distributions $P(X = x)$ by assigning the label x to n of the 2^r states, so that $P(X = x) \approx n/2^r$.

For example, say we save 3-bits to model a uniform distribution on $\{a, b, c\}$. There are $2^3 = 8$ possible bit strings of length 3. Assign the first three to a, the next three to b, and the

last two to c. This gives the approximated distribution with probabilities $(0.375, 0.375, 0.25)$. The same process using 5-bits gives the distribution:

$$(11/32, 11/32, 10/32) = (0.34375, 0.34375, 0.3125).$$

For any $\epsilon > 0$, we can approximate the true probabilities of $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ while ensuring a max error $< \epsilon$, by reserving more bits to model randomness.

2.2. **Reversibility.** We previously encoded probabilistic computations via stochastic matrices and states $v = \sum_I p_I |I\rangle$, with $p_I \geq 0$ and $\sum_I p_I = 1$, where the probability of state I is p_I. However, stochastic matrices in general are not invertible (e.g. the coin flip matrix was singular). This non-reversibility presents an issue because Schrodinger's equation and Stone's theorem modelled the time evolution of quantum states using families of *unitary* operators, which are invertible.

One way to think about a solution is by noticing that stochastic matrices preserve the ℓ_1-norm. But the more natural place to study quantum computation (from the quantum mechanical perspective) is that of a Hilbert space. This suggests we switch to the ℓ_2-norm, and use state vectors $v = \sum_I q_I |I\rangle$ with $\sum_I q_I^2 = 1$, where the probability of getting state I is q_I^2.

In terms of linear algebra, the matrices that preserve the ℓ_2-norm $\|x\| = \sqrt{\Sigma_I x_I^2}$ are the **orthogonal matrices**:

$$\begin{aligned}\mathcal{O}(n) &= \{A \in M_{m\times n}(\mathbb{R}) \mid \forall x \in \mathbb{R}^n : \|Ax\| = \|x\|\} \\ &= \{A \in M_{m\times n}(\mathbb{R}) \mid A^\top A = I\}.\end{aligned}$$

This switch is also beneficial in re-obtaining reversibility for probabilistic computations, since entries of orthogonal matrices are allowed to be negative.

Example 2.4 (Probabilistic coin flip). We can model a coin flip with the **Hadamard matrix**:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Given a starting state $|0\rangle$ or $|1\rangle$, the corresponding state vectors are $(1,0)^t$ or $(0,1)^t$, which H maps to $v = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. Computing the probabilities of obtaining $|0\rangle$ or $|1\rangle$, we get:

$$P(X=0) = P(X=1) = \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2},$$

which correctly models what we desire from a coin flip. The difference is that H is invertible, whereas the classical coin flip was not.

2.3. **Complexification.** We mentioned before that behaviors such as quantum interference can be best modelled mathematically by incorporating phases $e^{i\theta}$. We similarly saw that the solutions e^{itA} to Schrödinger's equation used complex numbers. This suggests switching to the complex counterparts of orthogonal matrices.

Recall that for $z \in \mathbb{C}$, its norm is $|z|^2 = z\bar{z}$. For $v = (z_I)_I \in \mathbb{C}^{2^k}$, we use the norm:

$$\|v\| = \sqrt{\Sigma_I |z_I|^2} = \sqrt{\Sigma_I z_I^\dagger z_I}.$$

Here $(-)^\dagger$ is the conjugate transpose operation. The matrices preserving the norm are now the **unitary matrices**:

$$\begin{aligned}\mathbf{U}(n) &= \{A \in M_{m\times n}(\mathbb{C}) \mid \forall v \in \mathbb{C}^n : |Av| = |v|\} \\ &= \{A \in M_{m\times n}(\mathbb{C}) \mid U^\dagger U = UU^\dagger = I\}.\end{aligned}$$

Our distribution vectors now look like $|v\rangle = \Sigma_I z_I |I\rangle$, where each $z_I \in \mathbb{C}$ is complex, their norms $\Sigma z_I^\dagger z_I = 1$ sum to 1, and the probability of obtaining state I is $|z_I|^2 = z_I^\dagger z_I$.

Proposition 2.5. Let $A \in \mathbf{U}(n)$. Then:

(1) Unitary matrices preserve norm: $\forall v, w \in \mathbb{C}^n : \langle Av \mid Aw\rangle = \langle v \mid w\rangle$.
(2) Unitary matrices have unit eigenvalues.
(3) Unitary matrices fix orthogonal complements of eigenvectors.
(4) Unitary matrices admit orthonormal bases of eigenvectors.

Remark 2.6 (Modelling continuous evolution). Last on our wishlist of upgrades is the desire for our probability vectors to depend continuously on a parameter t that model time evolution.

Discretely, this mean that given the matrix X of an admissible computation, with:

$$X|\psi_{t_0}\rangle = |\psi_{t_0+t_1}\rangle,$$

we wish to always be able to find admissible matrices Y, Z with $X = ZY$, such that:

$$Y|\psi_{t_0}\rangle = |\psi_{t_0+0.5t_1}\rangle \quad \text{and} \quad Z|\psi_{t_0+0.5t_1}\rangle = |\psi_{t_0+t_1}\rangle.$$

The following proposition shows that unitary matrices also give a good foundation to model time evolution.

Proposition 2.7. Unitary matrices have square roots: $\forall A \in \mathbf{U}(n) : \exists B \in \mathbf{U}(n) : B^2 = A$. Moreover, for each $\epsilon > 0$, unitary matrices have some m-th root ϵ-close to the identity.

$$\forall \epsilon > 0 : \forall A \in \mathbf{U}(n) : \exists m = m(\epsilon, n) \in \mathbb{N} : \exists C \in \mathbf{U}(n) : |C - I| < \epsilon \quad \text{and} \quad C^m = A.$$

Interlude on Hilbert Spaces

We mentioned that quantum states are represented either via wave functions or as vectors in a separable Hilbert space. We give definitions and results for the second perspective here.

Definition 2.8 (Hilbert Spaces). An **inner product space** $\mathcal{H}$ is a complex vector space equipped with an inner product $\langle - \mid - \rangle : \mathcal{H} \times \mathcal{H} \to \mathcal{H}$, which must satify:

- **Positive-definite**: $x \neq \vec{0}$ implies $\langle x \mid x \rangle > 0$.
- **Conjugate-symmetric**: $\forall x, y \in \mathcal{H} : \langle x \mid y \rangle = \overline{\langle y \mid x \rangle}$.
- **Sesquilinear**: $\langle - \mid - \rangle$ is linear in the second argument.

Call $\mathcal{H}$ a **Hilbert space** if its complete wrt the metric induced by the norm $\|x\| := \sqrt{\langle x \mid x \rangle}$. Note the last two axioms imply that $\langle \mid \rangle$ is conjugate linear in the first argument.

Example 2.9. $\mathbb{C}^n$ is a Hilbert space with the standard Hermitian inner product:

$$\langle x \mid y \rangle = x^\dagger y.$$

Theorem 2.10 (Inequalities in inner product spaces). Let V be equipped with an inner product. Then:

(1) Schwarz Inequality: $|\langle x \mid y \rangle| \leq \|x\| \|y\|$.

(2) Triangle Inequality: $\|x + y\| \leq \|x\| + \|y\|$.

Equality occurs in either iff x and y are nonnegative multiples.

Definition 2.11 (Orthogonality). Let V be an inner product space and let W, W' be subspaces.

- Call $x, y \in V$ **orthogonal** and write $x \perp y$ if $\langle x \mid y \rangle = 0$.
- Call $\{e_1, \ldots, e_n\} \subseteq V$ **orthonormal** if they are pairwise orthogonal and each vector has unit norm.
- Say $x \in V$ is **orthogonal** to W and write $x \perp W$ if $\langle w \mid x \rangle = 0$ for all $w \in W$.
- Call W and W' **orthogonal** and write $W \perp W'$ if each vector in one is orthogonal to the other subspace.
- The **orthogonal complement** of W, denoted $W^\perp$, is the set of all vectors orthogonal to W. It forms a subspace and decomposes $V = W \oplus W^\perp$ in finite dimensions.

Theorem 2.12 (Orthogonal Projections). Let V be an inner product space. Let W be the subspace spanned by some orthonormal set $\{e_1, \ldots, e_n\}$. For $v \in V$, the **orthogonal projection** $\text{proj}_W(v)$ of v onto W is the vector in W that minimizes the distance to v.

- The vector $v - \text{proj}_W(v)$ is orthogonal to W.
- The orthogonal projection is given by:

$$\text{proj}_W(v) = \sum_{i=1}^{n} \langle e_i \,|\, v \rangle e_i$$

Definition 2.13 (Bounded Maps). A linear map between Hilbert spaces $T : \mathcal{H} \to \mathcal{K}$ is **bounded** if $\exists r \in R$ such that $\|T(a)\| \leq r\|a\|$ for all $a \in \mathcal{H}$. One can show that linear operators are bounded iff they are continuous wrt the topologies induced by the norms on $\mathcal{H}$ and $\mathcal{K}$.

The smallest such r (or the infimum over all such $r \in \mathbb{R}_{\geq 0}$) is called the **operator norm** of T, denoted $\|T\|$. Intuitively, it represents the largest factor by which T lengthens vectors.

Notation (Categories of Hilbert spaces). The categories **Hilb** and **FHilb** consists of (finite-dimensional) Hilbert spaces and bounded linear maps. Linear maps between finite-dimensional Hilbert spaces are always bounded, so in fact **FHilb** is equivalent to **FVect**. We will use $\mathbf{Hilb}(\mathcal{H}, \mathcal{K})$ for hom-sets in **Hilb** (i.e. bounded linear maps) and $\mathbf{Vect}(\mathcal{H}, \mathcal{K})$ for hom-sets in **Vect** (i.e. arbitrary operators/linear maps).

Definition 2.14 (Adjoint Maps). The **adjoint** of $T : \mathcal{H} \to \mathcal{K}$ is the unique map $T^\dagger : \mathcal{K} \to \mathcal{H}$ satisfying:

$$\langle T^\dagger x \,|\, y \rangle_{\mathcal{H}} = \langle x \,|\, Ty \rangle_{\mathcal{K}} \quad \text{for all} \quad x, y \in \mathcal{H}.$$

Remark 2.15 (Existence of Adjoints). In **Hilb**, the fact that adjoints exist follows from the Riez Representation Theorem. However, unbounded maps need not have well-defined adjoints.

Remark 2.16 (A Categorical Perspective). The adjoint defines a contravariant endofunctor on **Hilb** satisfying $(T^\dagger)^\dagger = T$. Thus we have $1_{\mathcal{H}}^\dagger = 1_{\mathcal{H}}$ and $(T \circ S)^\dagger = S^\dagger \circ T^\dagger$.

In terms of Linear Algebra, adjoint maps correspond to conjugate transposed matrices.

Exercise 2.17. Show that if $T \in \text{End}(\mathbb{C}^n)$ has matrix A, them $T^\dagger$ has matrix $A^\dagger = \bar{A}^*$.

Fix a basis $\{e_i\}$ for $\mathbb{C}^n$ and let $A = (a_{ij})_{ij}$. Note that $a_{i,j} = \langle e_i \,|\, Te_j\rangle = \langle e_i \,|\, Ae_j\rangle$, so:

$$\langle e_i \,|\, T^\dagger e_j\rangle = \langle Te_i \,|\, e_j\rangle = \langle Ae_i \,|\, e_j\rangle = \overline{\langle e_j \,|\, Ae_i\rangle} = \overline{a_{ji}}.$$

Notation (Bras and Kets). Given $a \in \mathcal{H}$, its **ket** is the linear map $|a\rangle : \mathbb{C} \to \mathcal{H}$ with $1 \mapsto a$ tracing out the span of $a \in \mathcal{H}$. In many cases (specially when we assume all our vectors are normalized) we identify $a \in \mathcal{H}$ with its ket and write $|a\rangle \in \mathcal{H}$.

Its **bra** $\langle a|$ is the linear functional $\langle a \,|\, -\rangle : \mathcal{H} \to \mathbb{C}$. Writing $T = |a\rangle$, we have $T(y) = ya$ for each $y \in \mathbb{C}$. Noting that $\langle y \,|\, y'\rangle_{\mathbb{C}} = \bar{y}y'$, we see that:

$$\langle x \,|\, Ty\rangle_{\mathcal{H}} = \langle x \,|\, ya\rangle_{\mathcal{H}} = \langle x \,|\, a\rangle_{\mathcal{H}} y = \overline{\langle a \,|\, x\rangle_{\mathcal{H}}} y = \langle \langle a \,|\, x\rangle_{\mathcal{H}} \,|\, y\rangle_{\mathbb{C}},$$

which show that $\langle a|$ and $|a\rangle$ are adjoint.

The inner product "bra-kets" $\langle a \,|\, b\rangle \in \mathbb{C}$ of Hilbert spaces decompose into "bras" and "kets", in the sense that the endomorphism: $\langle b| \circ |a\rangle : \mathbb{C} \xrightarrow{|a\rangle} \mathcal{C} \xrightarrow{\langle b|} \mathbb{C}$ is given by scalar multiplication by $\langle b \,|\, a\rangle \in \mathbb{C}$. Using $\mathbb{C} \cong \mathbf{Hilb}(\mathbb{C}, \mathbb{C})$, we can identify $\langle b| \circ |a\rangle$ with $\langle b \,|\, a\rangle$.

Definition 2.18 (Dual Spaces). If $\mathcal{H}$ is a Hilbert space, its **algebraic dual space** is defined as the space of all linear functionals on $\mathcal{H}$, i.e. the vector space $\mathbf{Vect}(\mathcal{H}, \mathbb{C})$. The **continuous dual space** of $\mathcal{H}$ is the space of bounded linear functionals on $\mathcal{H}$, i.e. $\mathbf{Hilb}(\mathcal{H}, \mathbb{C})$. In finite dimensions these coincide, but more generally $\mathbf{Hilb}(\mathcal{H}, \mathbb{C})$ is a subspace of $\mathbf{Vect}(\mathcal{H}, \mathbb{C})$. Here, we deal only with continuous dual spaces, which we denote by $\mathcal{H}^*$.

By the Riesz Representation Theorem, each element of $\mathcal{H}^*$ has the form $\varphi_a := \langle a \,|\, -\rangle$. The dual space $\mathcal{H}^*$ is itself a Hilbert space with inner product given by:

$$\langle \varphi_a \,|\, \varphi_b\rangle = \langle a \,|\, b\rangle.$$

The map $i_{\mathcal{H}} : \mathcal{H} \to \mathcal{H}^*$ given by $a \mapsto \varphi_a = \langle a \,|\, -\rangle$ is invertible and anti-linear.

Definition 2.19 (Transpose Maps). Given $T : \mathcal{H} \to \mathcal{K}$, its **transpose** is the map $T^* : \mathcal{K}^* \to \mathcal{H}^*$ is the pull back along T, i.e. $T^*(f) = f \circ T$. The adjoint of T is simply its transpose after identifying $\mathcal{H}^*$ with $\mathcal{H}$.

$$T^\dagger = i_{\mathcal{H}}^{-1} \circ T^* \circ i_{\mathcal{K}}.$$

Definition 2.20. We define the following classes of operators for a bounded map T.

- **Hermitian** (or self-adjoint) operators satisfy $T^\dagger = T$,
- **idempotent** operators satisfy $T^2 = T$,
- **projections** are self-adjoint and idempotent,
- **isometries** satisfy $T^\dagger T = \mathrm{id}_H$,
- **unitary** operators satisfy $T^\dagger T = \mathrm{id}_H$ and $TT^\dagger = \mathrm{id}_K$,
- **partial isometries** are ones such that $T^\dagger T$ is a projection,
- **positive maps** decompose as $T = S^\dagger S$ for some $S \in \mathbf{Hilb}(\mathcal{H}, \mathcal{K})$.

Theorem 2.21 (Spectral Theorem). Let $\mathcal{H} \in \mathbf{FHilb}$. We call $T \in \mathrm{End}(\mathcal{H})$ **normal** if $TT^\dagger = T^\dagger T$. TFAE:

- T is normal.
- $\mathcal{H}$ admits an orthonormal basis of eigenvectors of T.
- Any matrix of T is unitarily diagonalizable.

Exercise 2.22 (2.1.20). Positive maps are Hermitian.

Let $T = S^\dagger S$. We have that for all $v \in \mathcal{H}$:

$$\langle S^\dagger S v \,|\, v\rangle = \langle Sv \,|\, Sv\rangle = \langle v \,|\, S^\dagger S v\rangle,$$

so $S^\dagger S$ is self-adjoint.

Definition 2.23 (Trace). The **trace** of a positive linear map $T : \mathcal{H} \to \mathcal{H}$ is:

$$\mathrm{Tr}(T) = \sum_i \langle e_i \,|\, T e_i\rangle,$$

for any orthonormal basis $\{e_i\}$ for $\mathcal{H}$ (assuming the sum converges).

Remark 2.24 (Characterizations of Trace). Using the isomorphism $\mathbf{Hilb}(H, H) \cong \mathbf{Hilb}(H^* \otimes H, \mathbb{C})$, the trace map is the canonical duality on $H^* \otimes H$, i.e. the evaluation map $f \otimes a \mapsto f(a)$.

When $\mathcal{H}$ is finite-dimensional, by the Spectral Theorem, we may pick an orthonormal eigenbasis for $\mathcal{H}$. In that case, note that $\langle e_i \,|\, T e_i\rangle = \lambda$, and the trace becomes the sum of the eigenvalues of T.

Definition 2.25 (Unitary Group). The **unitary group** of a Hilbert space $\mathcal{H}$ is the group of all unitary endomorphisms of $\mathcal{H}$ under composition. Equivalently, this is the group

consisting of invertible isometries whose inverse is given by their adjoint.

$$\begin{aligned}\mathbf{U}(\mathcal{H}) &= \{T \in \mathrm{End}(\mathcal{H}) :| \ \forall x \in \mathcal{H} : |Tx| = |x|\} \\ &= \{T \in \mathrm{End}(\mathcal{H}) \mid T^\dagger T = TT^\dagger = \mathrm{id}_\mathcal{H}\}.\end{aligned}$$

The unitary group is a real Lie group of dimension n^2 whose Lie algebra consists of the $n \times n$ skew-Hermitian matrices.

Remark 2.26 (Matrix Exponentials). The matrix exponential map $A \mapsto e^A$ maps a Lie algebra $\mathfrak{g}$ to its corresponding Lie group G. In particular, if you start with a smooth curve $U_t = e^{tA} \in G$ parametrized by $t \in (-\epsilon, \epsilon)$, notice that $U_0 = e^{0A} = \mathrm{id}$, so we may understand such curves as tangent vectors of G at id. Evaluating $\frac{d}{dt}(U_t)$ at $t = 0$ recovers A:

$$\left.\frac{d}{dt}U_t\right|_{t=0} = \left.\frac{d}{dt}e^{tA}\right|_{t=0} = \left.Ae^{tA}\right|_{t=0} = \left.AU_t\right|_{t=0} = A.$$

Slogan: "Lie algebras are tangent spaces of Lie groups at the identity".

Definition 2.27 (Unitary Lie Algebra). The **unitary Lie algebra** of a Hilbert space $\mathcal{H}$ as the set of skew-Hermitian endomorphisms:

$$\mathfrak{u}(\mathcal{H}) = \{X \in \mathrm{End}(\mathcal{H}) \mid X^\dagger = -X\}$$

equipped with the commutator bracket.

Remark 2.28 (Unitaries and Schrodinger). The association between $\mathbf{U}(\mathcal{H})$ and $\mathfrak{u}(\mathcal{H})$ was first hinted at through Schrodinger's equation and Stone's theorem, which established a correspondence $U_t = e^{itA}$ between smooth curves $U_t \in \mathbf{U}(\mathcal{H})$ and Hermitian operators A. In case you were wondering why A Hermitian and not skew-Hermitian: multiplying the (Hermitian) Hamiltonian A by the extra phase of i makes iA skew-Hermitian, as we'll see below.

Exercise 2.29. If A is Hermitian then iA is skew-Hermitian.

Let A be Hermitian. Then:

$$\langle x \,|\, iAy\rangle = i\langle Ax \,|\, y\rangle = \langle -iAx \,|\, y\rangle.$$

shows that $(iA)^\dagger = -iA$, making iA skew-Hermitian.

Remark 2.30 (Hermitian vs. skew-Hermitian). We prefer to frame things in terms of Hermitian matrices over skew-Hermitian matrices, because Hermitian matrices will correspond to our measurements/observables. Hermitian operators have real eigenvalues, while skew-Hermitian operators have imaginary eigenvalues.

Exercise 2.31. Eigenvalues of Hermitian operators are real.

Let A be Hermitian, so $A = A^\dagger$. For each eigenpairs $A\vec{v} = \lambda\vec{v}$:

$$\lambda\langle\vec{v},\vec{v}\rangle = \langle\vec{v},\lambda\vec{v}\rangle = \langle\vec{v},A\vec{v}\rangle = \langle A\vec{v},\vec{v}\rangle = \langle\lambda\vec{v},\vec{v}\rangle = \bar{\lambda}\langle\vec{v},\vec{v}\rangle.$$

Exercise 2.32. Eigenvalues of skew-Hermitian operators are purely imaginary.

Proof. Let X be skew-Hermitian ($X^\dagger = -X$) and $Xv = \lambda v$ be an eigenpair. Then:

$$\lambda\langle v \,|\, v\rangle = \langle v \,|\, Xv\rangle = \langle -Xv \,|\, v\rangle = -\langle \lambda v \,|\, v\rangle = -\bar{\lambda}\langle v \,|\, v\rangle.$$

Dividing by $\langle v \,|\, v\rangle \neq 0$ shows $\lambda = -\bar{\lambda}$, implying $\mathrm{Re}(\lambda) = 0$. □

To make this more formal, we consider smooth curves in $U(t) \in \mathbf{U}(\mathcal{H})$ with $U(0) = \mathrm{id}$. Notice that for $v, w \in \mathcal{H}$:

$$\begin{aligned} 0 &= \frac{d}{dt}\bigg|_{t=0} \langle v \,|\, w\rangle \\ &= \frac{d}{dt}\bigg|_{t=0} \langle v \,|\, U(t)U(t)^\dagger w\rangle \\ &= \langle v \,|\, U'(0)w\rangle + \langle v \,|\, U'(0)w\rangle, \end{aligned}$$

so $\langle v \,|\, U'(0)w\rangle = \langle v \,|\, -U'(0)w\rangle$. Since this holds for all vectors, $U'(0)^\dagger = -U'(0)$.

Exercise 2.33 (2.3.12). Show that skew-Hermitian matrices are diagonalizable.

Proof. If X is skew-Hermitian, then $XX^\dagger = X^\dagger X$ because:

$$X^\dagger X = (-X)X = -X^2 \quad \text{and} \quad XX^\dagger = -X^2.$$

Hence X is normal, so unitarily diagonalizable by the Spectral Theorem. □

3. Quantum States

3.1. **Qubits.** The classical bit had two possible states, 0 and 1. The quantum version of a bit will allow for "probabilistic mixtures" of these two classical states. For reasons explained in the previous section, these will live in complex Hilbert spaces, and probabilities will be obtained using the corresponding ℓ_2-norm.

Definition 3.1 (Qubits). A **qubit** $|\psi\rangle$ is a unit vector in a finite-dimensional Hilbert space $\mathcal{H}$ isomorphic to $\mathbb{C}^2$. Given a basis $|0\rangle, |1\rangle \in \mathcal{H}$, these can be expressed as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \in \mathcal{H} \quad \text{or via their coordinate vectors} \quad \begin{matrix}|0\rangle \\ |1\rangle\end{matrix} \begin{bmatrix}\alpha \\ \beta\end{bmatrix} \in \mathbb{C}^2.$$

The constraint $\langle\psi \,|\, \psi\rangle = 1$ is equivalent to $|\alpha|^2 + |\beta|^2 = 1$.

Definition 3.2 (N-Qubits). An **n-qubit** is a unit vector in $\mathcal{H} \cong (\mathbb{C}^2)^{\otimes n} \cong \mathbb{C}^{2^n}$. Similar to classical computation via linear algebra, we denote bases of $\mathcal{H}$ as:

$$\{|I\rangle = |x_1\rangle \otimes \cdots \otimes |x_{2^n}\rangle \mid I = x_1 \cdots x_n \in \mathbb{F}_2^n\} \qquad \text{or} \qquad \{|k\rangle \mid 0 \leq k \leq 2^n - 1\},$$

both ordered lexicographically. A general n-qubit thus has form:

$$|\psi\rangle = \sum_{I \in \mathbb{F}_2^n} a_I |I\rangle \quad \text{or} \quad |\psi\rangle = \sum_{k=0}^{2^n-1} a_k |k\rangle$$

We call the coefficient a_I the **probability amplitude** of states $|I\rangle$.

Definition 3.3 (Joint Systems). Given two quantum systems with state spaces $\mathcal{H}$ and $\mathcal{K}$, it is a postulate of quantum mechanics that their joint system has state space $H \otimes K$. For $|a\rangle \in \mathcal{H}$ and $|b\rangle \in \mathcal{K}$, we write $|ab\rangle = |a\rangle \otimes |b\rangle$. Simple tensors in $\mathcal{H} \otimes \mathcal{K}$ are called **product states** or **separable states**. Non-simple tensors are called **entangled states**.

3.2. **Bases for qubits and 2-qubits.** We describe important bases used to describe 1-qubit and 2-qubit systems. For single qubits, the following are three widely used bases.

Definition 3.4 (X,Y,Z bases). Fix a computational basis $|0\rangle, |1\rangle$ for $\mathcal{H} \cong \mathbb{C}^2$. We call $|0\rangle, |1\rangle$ the **Z-basis** of $\mathcal{H}$. The **X-basis** $|+\rangle, |-\rangle$ and Y**-basis** $|R\rangle, |L\rangle$ are defined as:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad |R\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle),$$
$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \qquad |L\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle).$$

For two-qubits, the following orthonormal basis is very important.

Definition 3.5 (Bell Basis). The **Bell basis** for a 2-qubit system $\mathcal{H} \cong \mathbb{C}^2 \otimes \mathbb{C}^2$ is:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle),$$
$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle),$$
$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle),$$
$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

It is useful to write the standard states in terms of the Bell states:

$$|00\rangle = \frac{1}{\sqrt{2}}\left(|\Phi^+\rangle + |\Phi^-\rangle\right),$$
$$|01\rangle = \frac{1}{\sqrt{2}}\left(|\Psi^+\rangle + |\Psi^-\rangle\right),$$
$$|10\rangle = \frac{1}{\sqrt{2}}\left(|\Psi^+\rangle - |\Psi^-\rangle\right),$$
$$|11\rangle = \frac{1}{\sqrt{2}}\left(|\Phi^+\rangle - |\Phi^-\rangle\right).$$

3.3. **The Bloch Sphere.** The Bloch sphere is a geometric representation of the state space of a single qubit as a 2-sphere. It is useful in visualizing how operators act on single qubits.

Let $|\psi\rangle \in \mathbb{C}^2$ be a qubit. Given a basis $|0\rangle, |1\rangle$ of $\mathbb{C}^2$, the state $|\psi\rangle$ is determined by four real parameters coming from the two complex coefficients. In polar form:

$$|\psi\rangle = r_1 e^{i\delta}|0\rangle + r_2 e^{i\delta'}|1\rangle.$$

where $r_i \in \mathbb{R}_{\geq 0}$ and $0 \leq \delta, \delta' < 2\pi$. The normalization constraint $\langle\psi \,|\, \psi\rangle = 1$ is equivalent to $|r_1|^2 + |r_2|^2 = 1$, which removes a degree of freedom. Writing $r_1 = \cos(\theta/2)$ with $0 \leq \theta \leq \pi$

allows us to write $|\psi\rangle$ as:

$$|\psi\rangle = e^{i\delta}\cos(\theta/2)|0\rangle + e^{i\delta'}\sin(\theta/2)|1\rangle.$$

We also write $\delta' = \delta + \varphi$ to emphasize the difference in the phases over their values.

$$|\psi\rangle = e^{i\delta}\cos(\theta/2)|0\rangle + e^{i(\delta+\varphi)}\sin(\theta/2)|1\rangle.$$

This gives the **Hopf coordinates** of $|\psi\rangle$:

$$\begin{bmatrix} e^{i\delta}\cos(\theta/2) \\ e^{i(\delta+\varphi)}\sin(\theta/2) \end{bmatrix}.$$

Now notice that the shared phase $e^{i\delta}$ does not affect the probability amplitudes of $|0\rangle, |1\rangle$ because $|e^{i\delta}|^2 = 1$. In other words, pure states that differ by a phase specify the same probability distributions on $|0\rangle, |1\rangle$. WLOG, we can multiply out the shared phase term to obtain a unique representation:

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle,$$

in terms of two real parameters $0 \leq \theta \leq \pi$ and $0 \leq \varphi < 2\pi$.

The **Bloch vector** $\vec{a}$ of $|\psi\rangle$ can be obtained by using θ and φ to specify spherical coordinates in the unit sphere of $\mathbb{R}^3$.

$$\vec{a} = \begin{bmatrix} \sin\theta\cos\varphi \\ \sin\theta\sin\varphi \\ \cos\theta \end{bmatrix}.$$

As θ and φ vary, we cover the entire sphere, which we call the **Bloch sphere**. Notice that:

- The two points on the z-axis correspond to the Z-basis $|0\rangle, |1\rangle$.
- The two points on the x-axis correspond to the X-basis:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

- The two points on the y-axis correspond to the Y-basis:

$$|R\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \qquad |L\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle).$$

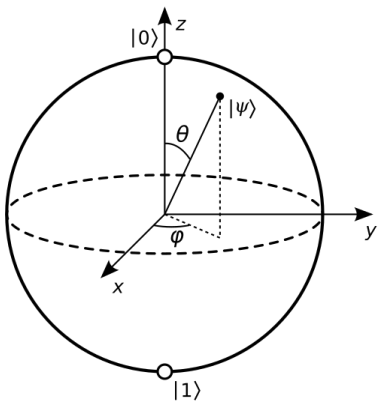


FIGURE 1. Bloch Sphere

4. Purity and Measurements

4.1. **Measuring Pure States.** The notion of taking a measurement in quantum theory is connected to Hermitian operators. We've seen that these have real eigenvalues, which will be our *observables* after measurement. The simplest notion of measurement a a *projection-valued measure*. Recall that projections are idempotent and Hermitian.

Definition 4.1. A finite family of linear maps $f_i : \mathcal{H} \to \mathcal{H}$ is **complete** if $\Sigma_i f_i = \mathrm{id}_{\mathcal{H}}$. A family of such maps is **orthogonal** if $i \neq j$ implies $f_i f_j = 0$. A **projection-valued measure (PVM)** on $\mathcal{H}$ is a finite family of projections $\{p_i\}$ that are complete and orthogonal. A **non-degenerate** PVM is one where $\mathrm{Tr}(p_i) = 1$ for each i.

Lemma 4.2. In $\mathcal{H} \in$ **FHilb**, nondegenerate PVMs are in correspondence with orthonormal bases, up to phase.

Proof. For $|i\rangle$ ON, defining $p_i = |i\rangle\langle i|$ gives a nondegenerate PVM of projections onto the basis vectors. Conversely, since projections p have eigenvalues 1, nondegeneracy $\mathrm{Tr}(p) = 1$ implies that p has rank one. Thus, there is a suitable $|i\rangle$ for which $p = |i\rangle\langle i|$, which becomes unique up to multiplication by $e^{i\theta}$. □

Born's rule defines the probabilistic distribution given by PVMs.

Definition 4.3. Given a PVM $\{p_i\}$ the **probability of outcome i** on a system in a normalized state $a \in \mathcal{H}$ is:

$$\langle a \,|p_i|\, a\rangle := \langle a \,|\, p_i a\rangle = \langle p_i a \,|\, a\rangle.$$

Notice that:

$$\Sigma_i \langle a \,|p_i|\, a\rangle = \langle a \,|\Sigma_i p_i|\, a\rangle = \langle a \,|\, a\rangle = 1.$$

After measurement, the new (possibly unnormalized) state is postulated to be $p_i(a)$.

4.2. **Measuring Mixed States.** Many times in quantum theory, you will be given a system whose state is unknown, but can be described probabilistically. In these cases, we can't describe states as elements of H, so we need a more general notion.

Definition 4.4 (Density Matrix). A **density matrix** on H is a positive map $m : \mathcal{H} \to \mathcal{H}$. A **normalized** density matrix has $\mathrm{Tr}(m) = 1$. A density matrix is **pure** when $m = |a\rangle\langle a|$ and **mixed** otherwise.

Recall that positive maps decompose as $m = g^\dagger g$, so density matrices are Hermitian. Every pure state $|a\rangle \in \mathcal{H}$ gives rise to a canonical pure density matrix $m = |a\rangle\langle a| = \langle a|^\dagger \langle a|$.

Definition 4.5. Let $H \in \mathbf{FHilb}$. The **maximally mixed state** is the density matrix:

$$\frac{1}{\dim \mathcal{H}} \mathrm{id}_{\mathcal{H}}\,.$$

Definition 4.6. Given density matrices $m, n : \mathcal{H} \to \mathcal{H}$, a **convex combination** is of form $sm + tn$ with s, t non-negative integers with $s + t = 1$.

It can be proved that convex combinations of density matrices are density matrices. Physically, they describe the state of systems produced by machines that output state m with probability s and state n with probability t. In finite-dimension, *every* mixed state is a convex combination of pure states (not uniquely).

We also need to generalize PVMs to deal with density matrices.

Definition 4.7. A **positive operator valued measure (POVM)** on $\mathcal{H}$ is a family of positive maps $f_i : \mathcal{H} \to \mathcal{H}$ satisfying $\Sigma_i f_i = \mathrm{id}_{\mathcal{H}}$.

We can see that PVMs are POVMs with only pure states by setting $f_i = p_i$.

Definition 4.8 (Born's Rule for POVMs). Given a POVM f_i, the **probability of outcome i** on a system with normalized density matrix m is $\mathrm{Tr}(f_i m)$.

Definition 4.9 (Partial Trace). Given $\mathcal{H}, \mathcal{K} \in \mathbf{Hilb}$, there is a unique linear map $Tr_{\mathcal{K}} \in \mathrm{End}_{\mathbf{Hilb}}(\mathcal{H} \otimes \mathcal{K})$ characterized by $\mathrm{Tr}_{\mathcal{K}}(m \otimes n) = \mathrm{Tr}(n)m$, called the **partial trace** over $\mathcal{K}$.Given an orthonormal basis $|i\rangle$ for K,

$$\mathrm{Tr}_{\mathcal{K}}(f) = \Sigma_i (1_{\mathcal{H}} \otimes \langle i|) \circ f \circ (1_{\mathcal{H}} \otimes |i\rangle).$$

The partial trace over $\mathcal{H}$ is defined similarly.

Definition 4.10. A pure state $a \in \mathcal{H} \otimes \mathcal{K}$ is **maximally entangled** if tracing out $|a\rangle\langle a|$ over $\mathcal{H}$ or K gives a maximally mixed state, i.e.

$$\mathrm{Tr}_{\mathcal{H}}(|a\rangle\langle a|) = s\,\mathrm{id}_{\mathcal{K}} \qquad \mathrm{Tr}_{\mathcal{K}}(|a\rangle\langle a|) = t\,\mathrm{id}_{\mathcal{H}}\,.$$

In particular, when a is normalized, $s = 1/\dim \mathcal{H}$ and $t = 1/\dim \mathcal{K}$.

Interlude on Fourier Transforms

Much of the content from this section comes from Boggess and Narcowich's *A First Course on Wavelets with Fourier Analysis*.

Introduction. The basic goal of Fourier analysis is taking signals (input units = time) and decompose/transform them into functions of frequency. The units used are:

- frequency: # events/cycles/reps per unit of time.
- period (1/frequency): amount of time between each event/cycle.

The basic building blocks are $\sin(kt), \cos(kt)$ or $e^{ikt} = \cos(kt) + i\sin(kt)$ which vibrate at frequencies of k times per 2π time intervals.

Common applications include cleaning signals (deleting/truncating high frequency terms, which we consider "noise"), compressing data (get a good and simple approximation by eliminating coefficients below a threshold), and solving PDEs like the heat equation.

In applications, even if we know there are underlying continuous functions $f(t)$ modeling the data, computers often (1) only consider $f(t)$ at a finite time interval $[a, b]$ of interest, and (2) only simulate $f(t)$ by sampling it at N points $(t_i, f(t_i))_{i=0}^{N-1}$, where N is large enough but finite. We usually choose $N = 2^n$ to work nicely in classical computers, and choose the sampling points to be evenly spaced $t_k = a + \frac{k}{N}|b - a|$ for $k = 0, \ldots, N-1$. Finally, to simplify calculations, we usually transform $[a, b]$ to $[0, 2\pi]$, $[-\pi, \pi]$, or $[-1, 1]$.

Fourier Series. The space $L^2([0, 2\pi])$ consists of functions:

$$f : [0, 2\pi] \to \mathbb{C} \quad \text{satisfying} \quad \int_0^{2\pi} |f(t)|^2 dt < \infty.$$

It is an inner product space via $\langle f \,|\, g \rangle = \frac{1}{2\pi} \int_0^{2\pi} \overline{f(t)} g(t) dt$. The 2π factor is optional: we divide it out to normalize the space and make e^{ikt} have unit length. **For convenience, we will use the inner product notation for functions, regardless of square-integrability**.

The functions $\{e^{ikt}\}$ form an orthonormal (Schauder) basis for $L^2[0, 2\pi]$. This allows us to take the inner products $\langle e^{ikt} \,|\, f \rangle$ (the component of f along e^{ikt}) and approximate f with $\Sigma_k \langle e^{ikt} \,|\, f \rangle e^{ikt}$.

The discrete analog of L^2 is the space of square-summable sequences:

$$\ell^2 = \left\{ x = (x_n) : \mathbb{Z} \to \mathbb{C} \mid \Sigma_{-\infty}^{\infty} |x_n|^2 < \infty \right\} \quad \text{with} \quad \langle x \,|\, y \rangle = \Sigma_{-\infty}^{\infty} \overline{x_n} y_n.$$

Definition 4.11. The **Fourier series** of $f : [0, 2\pi] \to \mathbb{R}$ is $S_f = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikt}$, where the **Fourier coefficients** α_k are defined as $\alpha_k = \langle e^{ikt} \mid f \rangle = \frac{1}{2\pi} \int_0^{2\pi} f(t) e^{-ikt} dt$.

- Since $\alpha_{-k} = \bar{\alpha_k}$ and $z + \bar{z} = 2\,\mathrm{Re}(z)$, we get $S_f = \alpha_0 + 2\,\mathrm{Re}\left(\sum_{k=1}^{\infty} \alpha_k e^{ikt}\right)$.
- Using Euler's formula and the above, we can also write the Fourier series as:

$$S_f = a_0 + \sum_{k=1}^{\infty} a_k \cos(kt) + b_k \sin(kt)$$

  where $a_0 = \alpha_0$, $a_n = \frac{1}{\pi} \int_0^{2\pi} f(t) \cos(kt) dt$, and $b_n = \frac{1}{\pi} \int_0^{2\pi} f(t) \sin(kt) dt$.
- The complex and real Fourier coefficients are related via $\alpha_k = \frac{1}{2}(a_n - i b_n)$.

We list a couple key results next.

Theorem 4.12. Let $f : [0, 2\pi] \to \mathbb{R}$.

- If $f(t) = \sum_{k=-\infty}^{\infty} \alpha'_k e^{ikt}$ on $[0, 2\pi]$, then α'_k are precisely the Fourier coefficients of f.
- **(Riemann-Lebesgue Lemma)** If f is integrable, $\alpha_k \to 0$ as $k \to \infty$.
- If f is continuous, $S_f \to f$ pointwise at each point where the derivative is defined.
- If f is left and right differentiable at t (e.g f is piecewise differentiable with a jump discontinuity at t), then its Fourier series converges to the average $\frac{1}{2}[f(t_-) + f(t_+)]$ of its left and right limits.

Theorem 4.13. Let $f \in L^2[0, 2\pi]$.

- The partial sum $S_{f,n} = \sum_{k=-n}^{n} \alpha_k e^{ikt}$ is the projection of f onto $\mathrm{span}\{e^{ikt}\}_{k=-n}^{n}$.
- $S_{f,n} \to f$ in L^2-norm as $n \to \infty$.
- **(Parseval's Equation)** If $f = S_f$ then $\|f\|^2 = \Sigma_{-\infty}^{\infty} |\alpha_k|^2$. Moreover, if $f, g \in L^2[0, 2\pi]$ then $\langle f \mid g \rangle = \Sigma_{-\infty}^{\infty} \bar{\alpha_k} \beta_k$.
- **(Bessel's Inequality)** $\Sigma_{-n}^{n} |\alpha_k|^2 \leq \|f\|^2$.

Remark 4.14. The L^2-norm of f is often interpreted as energy. In that case, Parseval's equation states that the energy of a signal is the sum of the energy at each frequency component.

Discrete Fourier Transforms. Given a function $g : [0, 2\pi] \to \mathbb{R}$, approximating $\frac{1}{2\pi}\int_0^{2\pi} f(t)dt$ using the trapezoidal rule with uniform step sizes $\{t_j = \frac{2\pi j}{n}\}_{j=0}^{n}$ gives:

$$\frac{1}{2\pi}\int_0^{2\pi} g(t)dt \approx \frac{1}{n}\left(\frac{g(t_0)}{2} + \frac{g(t_n)}{2} + \sum_{j=1}^{n-1} g(t_k)\right).$$

If we assume g is 2π-periodic this simplifies to $\frac{1}{n}\sum_{j=0}^{n-1} g(t_j)$. Applying this to the formulas for the Fourier coefficients of f gives:

$$\alpha_k = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-ikt}dt \approx \frac{1}{n}\sum_{j=0}^{n-1} f(t_j)\omega^{-jk},$$

where $\omega = e^{\frac{2\pi i}{n}}$ is the n-th root of unity.

Remark 4.15. Since α_k are not n-periodic, the above is not a good approximation for $k \geq n$. Furthermore, the trapezoidal rule only gives good approximations if k is small relative to n: the higher the frequency, the faster the values fluctuate relative to a fixed step size.

Definition 4.16. Let $\mathcal{S}_n$ be the vector space of n-periodic complex sequences $y : \mathbb{Z} \to \mathbb{C}$. Let $y = (y_j)_{-\infty}^{\infty} \in \mathcal{S}_n$. The **discrete Fourier transform** of y is the sequence $(\mathcal{F}_n\{y\})_k = \widehat{y_k}$:

$$\widehat{y_k} = \sum_{j=0}^{n-1} y_j \omega_n^{-jk}.$$

The **inverse discrete Fourier transform** of $\hat{y}$ is given by:

$$y_j = \frac{1}{n}\sum_{k=0}^{n-1} \widehat{y_k}\omega^{jk}.$$

Theorem 4.17 (Matrix DFT). Enconding sequences as vectors $y = (y_0, y_1, \ldots, y_{n-1})^T$, the computation of the DFT is equivalent to matrix multiplication $\widehat{y} = \mathbf{DFT}_n^{\dagger} y$, where $\mathbf{DFT}_n$ is the Vandermonde matrix:

$$\mathbf{DFT}_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & & \vdots & & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}$$

The inverse DFT is equivalent to multiplying $\hat{y} = (\widehat{y_0}, \ldots, \widehat{y_{n-1}})^T$ by $\frac{1}{n}\mathbf{DFT}_n$. Note that **DFT** is symmetric, so the conjugate transpose simply conjugates the entries.

Theorem 4.18 (Properties of DFT). The DFT and $\mathcal{S}_n$ satisfy the following properties:

- **Linearity:** The DFT defines a linear operator $\mathcal{F} : \mathcal{S}_n \to \mathcal{S}_n$.
- **Unitary DFT:** The DFT and inverse DFT matrices satisfy:

$$\frac{\mathbf{DFT}_n}{\sqrt{n}} \circ \frac{\mathbf{DFT}_n^\dagger}{\sqrt{n}} = \mathrm{id}_n \,.$$

- **Translations to Phases:** If $z_k = y_{k+1}$ then $\mathcal{F}[z]_k = \omega^k \mathcal{F}[y]_k$.
- **Convolutions:** If $y, z \in \mathcal{S}_n$ then their **convolution** $y * z \in \mathcal{S}_n$, where:

$$[y * z]_k = \sum_{j=0}^{n-1} y_j z_{k-j}.$$

- **Convolutions to Products:** $\mathcal{F}[y * z]_k = \mathcal{F}[y]_k \mathcal{F}[z]_k$.
- **Conjugation and Time Inversion:** $\overline{\mathcal{F}[y]_k} = \mathcal{F}[y]_{n-k}$.

Fast DFT. Since the DFT can be implemented through matrix multiplication, it takes n^2 multiplications to perform. The fast Fourier transform (FFT) reduces this to $\approx 5n \log_2 n$. We will need n to be a power of two (padding with zeros if necessary). A formal proof would proceed by induction, but we just focus on one iteration and write $n = 2N$ and $W = \omega^2$.

Let $y \in \mathcal{S}_{2N}$ and split $y_{even} = (y_0, y_2, \ldots, y_{2N-2})$ and $y_{odd} = (y_1, y_3, \ldots, y_{2N-1}) \in \mathcal{S}_N$.

$$\begin{aligned}
\mathcal{F}_{2N}[y]_k &= \sum_{j=0}^{2N-1} y_j \omega^{-jk} \\
&= \sum_{j=0}^{N-1} y_{2j} \omega^{-2jk} + \sum_{j=0}^{N-1} y_{2j+1} \omega^{-(2j+1)k} \\
&= \sum_{j=0}^{N-1} y_{2j} W^{-jk} + \omega^{-k} \sum_{j=0}^{N-1} y_{2j+1} W^{-jk} \\
&= \mathcal{F}_N[y_{even}]_k + \omega^{-k} \mathcal{F}_N[y_{odd}].
\end{aligned}$$

Note that $\mathcal{F}_N[y_{even}], \mathcal{F}_N[y_{odd}] \in \mathcal{S}_N$. Use this plus the fact that $\omega^{-(k+N)} = -\omega^{-k}$ to substitute $k \mapsto k + N$ in the equations above to get:

$$\mathcal{F}_{2N}[y]_{k+N} = \mathcal{F}_N[y_{even}]_k - \omega^{-k}\mathcal{F}_N[y_{odd}].$$

This cuts our required computations by half at each iteration.
In terms of matrix multiplication, the decomposition above looks like:

$$\mathcal{F}_{2N}[y] = \mathbf{DFT}^\dagger_{2N}\, y = \begin{bmatrix} I_N & D_N^\dagger \\ I_N & -D_N^\dagger \end{bmatrix} \begin{bmatrix} \mathbf{DFT}^\dagger_N & 0 \\ 0 & \mathbf{DFT}^\dagger_N \end{bmatrix} \begin{bmatrix} y_{even} \\ y_{odd} \end{bmatrix},$$

where $D_N = \mathrm{diag}(1, \omega, \omega^2, \ldots, \omega^{N-1})$.

Let comp(L) be the number of required multiplications to perform the DFT this way for $n = 2^L$. The block DFT matrix requires $\sim 2\,\mathrm{comp(L\text{-}1)}$ multiplications, while the second matrix requires 2^{L-1}. Plugging in values, we get $\mathrm{comp}(1) \approx 1$, $\mathrm{comp}(2) = 2^1 \times 2$, $\mathrm{comp}(3) = 2^2 \times 3$, and generally $\mathrm{comp(L)} \approx 2^{L-1} \times L \sim n \log_2 n$.

Generalizations. The theory of the Fourier transform can be generalized to certain subclasses of abelian groups, which importantly include the finite abelian groups, the integers, the circle group, and the real numbers.

Like any good generalization, this theory will recover Fourier series, DFT, and the continuous FT when applying it to specific groups. We won't build this theory from the ground up here, since much of that material won't be of use for our purposes. However, I'll include some words on the tools needed below:

- We need our group to have a **topology** to discuss continuity or integrals. This is commonly done via locally compact abelian groups. The topology on $\mathbb{Z}$ and finite abelian groups will be discrete. On $\mathbb{R}$ or the circle group $\mathbb{T}$, we use the standard topologies.
- We need a good notion of **integration**. This is commonly done via the Haar measure of a locally compact abelian group. This measure satisfies desirable properties on Borel sets (translation invariance, inner/outer regularity, finiteness on compacts) and is unique up to scaling in that respect. For discrete groups, the measure will be the

counting measure. For $\mathbb{R}$ it's the Lebesgue measure. For $\mathbb{T}$, we parametrize the circle $[0, 2\pi] \to \mathbb{T}$ with $\theta \mapsto (\cos\theta, \sin\theta)$ and pull back the Lebesgue measure.

- We need a **basis** of functions on our group G that serve as our **frequencies**. This is done by considering the Pontryagin dual $\widehat{G}$ of G. For our purposes, define $\widehat{G}$ as the group of characters $G \to \mathbb{T} \subseteq \mathbb{C}^\times$ of G. The characters of G are precisely its one-dimensional unitary representations. These generalize e^{ikt} in several ways, e.g. by (1) turning addition in G into multiplication of phases, (2) having $|\chi(g)| = |e^{ikt}| = 1$, (3) being are class functions, which come with a natural inner product:

$$\langle \chi_1 \,|\, \chi_2 \rangle = \frac{1}{|G|} \sum_{g \in G} \chi_1(g)\overline{\chi_2(g)},$$

for finite groups G, and (4) forming an orthonormal basis for the space of class functions (which are just functions $G \to \mathbb{C}^\times$ in the abelian cases). We have:

$$\widehat{\mathbb{Z}/n\mathbb{Z}} \cong \mathbb{Z}/n\mathbb{Z}, \quad \widehat{\mathbb{R}} \cong \mathbb{R}, \quad \widehat{\mathbb{Z}} \cong \mathbb{T}, \quad \widehat{\mathbb{T}} \cong \mathbb{Z}.$$

For finite abelian groups $G = \prod_i \mathbb{Z}_{n_i}$, Pontryagin duals and characters all decompose nicely into components, e.g. $\widehat{G} = \prod_i \widehat{\mathbb{Z}}_{n_i}$ and $\chi(x_1, \ldots, x_\ell) = \chi_1(x_1) \ldots \chi_\ell(x_\ell)$.

Definition 4.19. Let G be a locally compact abelian group with Haar measure μ and Pontryagrin dual $\widehat{G}$. For each $f \in L^1(G)$, its **Fourier transform** is a function on $\widehat{G}$:

$$\hat{f}(\chi) = \int_G f(x)\overline{\chi(x)} d\mu(x).$$

Theorem 4.20. For each Haar measure μ on G there is a unique Haar measure ν on $\widehat{G}$ such that whenever $f \in L^1(G)$ and $\hat{f} \in L^1(\widehat{G})$ then:

$$f(x) =_{\mu \text{ a.e.}} \int_{\widehat{G}} \hat{f}(\chi)\chi(x) d\nu(\chi).$$

Definition 4.21. The **inverse Fourier transform** of $g \in L^1(\widehat{G})$:

$$\check{g}(x) = \int_{\widehat{G}} g(\chi)\chi(x) d\nu(\chi).$$

Example 4.22. We recover all our Fourier transforms as follows:

- $\mathcal{F} : (G = \mathbb{R}) \to (\widehat{G} \cong \mathbb{R})$ is the continuous Fourier transform.

- $\mathcal{F} : (G = \mathbb{Z}_n) \to (\widehat{G} \cong \mathbb{Z}_n)$ give the DFT.
- $\mathcal{F} : (G = \mathbb{T}) \to (\widehat{G} \cong \mathbb{Z})$ give Fourier series.
- $\mathcal{F} : (G = \mathbb{Z}) \to (\widehat{G} \cong \mathbb{T})$ give discrete-time Fourier transforms.

Example 4.23 (DFT). Let's recover the DFT from $G = \mathbb{Z}_n$. Equip the functions $\mathbb{Z}_n \to \mathbb{C}$ with the inner product $\langle f \,|\, g \rangle = \frac{1}{n} \sum_{k=0}^{n-1} \overline{f(k)} g(k)$. The characters of $\mathbb{Z}_n$ form an orthonormal basis, and are of the form $\chi_m(k) = \omega^{mk}$ for $\omega = e^{\frac{2\pi i}{n}}$. We identify $\widehat{\mathbb{Z}}_n \cong \mathbb{Z}_n$ by $\chi_m \mapsto m$ to make $\hat{f}$ a function on $\mathbb{Z}_n$. The Fourier transform and inverse Fourier transforms are given by:

$$\hat{f}(k) = \langle \chi_m \,|\, f \rangle = \frac{1}{n} \sum_{k=0}^{n-1} f(k) \omega^{-km} \quad \text{and} \quad f(k) = \sum_{m=0}^{n-1} \hat{f}(m) \omega^{km}.$$

Up to scalar multiple (based on our choice of inner product), this precisely the DFT.

Example 4.24 ($\mathbb{Z}_2$ DFT). In the case when $G = \mathbb{Z}_2$, the nontrivial characters are $\chi_0(x) = 1$ and $\chi_1(x) = (-1)^x$. Use the natural inner product:

$$\langle f \,|\, g \rangle = \frac{1}{2} \sum_{x \in \mathbb{Z}_2} \overline{f(x)} g(x).$$

Then the Fourier transform of $f : \mathbb{Z}_2 \to \mathbb{C}$ with $f(0) = a$ and $f(1) = b$ has:

$$\hat{f}(0) = \langle \chi_0 \,|\, f \rangle = \frac{1}{2}(a + b) \quad \text{and} \quad \hat{f}(1) = \langle \chi_1 \,|\, f \rangle = \frac{1}{2}(a - b).$$

The inverse Fourier transform of $g : \mathbb{Z}_2 \to \mathbb{C}$ with $g(0) = c$ and $g(1) = d$ is:

$$\check{g}(0) = (-1)^{0 \cdot 0} g(0) + (-1)^{0 \cdot 1} g(1) = c + d$$
$$\check{g}(1) = (-1)^{1 \cdot 0} g(0) + (-1)^{1 \cdot 1} g(1) = c - d.$$

Informally, $\hat{f}$ is taking a function f on a single bit and decomposing it into the frequencies $\chi_0 = 1$ (how much is f like a constant function?) and $\chi_1 = (-1)^-$ (how much is f like an alternating function?). The answers to these questions are fully determined by the two values $f(0), f(1)$ it takes on. The degrees to which f is constant or changing are exactly inversely proportional, so both $\hat{f}(0)$ and $\hat{f}(1)$ are weighed equally. The best constant function to model f is the average of its two values, while the "changing" part of f is modeled by the difference in the values.

The unitary version of this DFT uses the inner product:

$$\langle f \,|\, g \rangle = \frac{1}{\sqrt{2}} \sum_{x \in \mathbb{Z}_2} \overline{f(x)} g(x).$$

It splits the normalization factor $1/|G| = 1/2$ equally between $\mathcal{F}$ and $\mathcal{F}^{-1}$. The Fourier transform of $f : \mathbb{Z}_2 \to \mathbb{C}$ with $f(0) = a$ and $f(1) = b$ is:

$$\hat{f}(0) = \langle \chi_0 \,|\, f \rangle = \frac{1}{\sqrt{2}} (a + b) \quad \text{and} \quad \hat{f}(1) = \langle \chi_1 \,|\, f \rangle = \frac{1}{\sqrt{2}} (a - b).$$

In matrix form, $(\hat{f}(0), \hat{f}(1))^T = \mathbf{DFT}_2 (f(0), f(1))^T$, where:

$$\mathbf{DFT}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Example 4.25 ($\mathbb{Z}_2^n$ DFT). The characters of $G = \mathbb{Z}_2^n$ are $\chi_\omega(x) = (-1)^{\langle \omega \,|\, x \rangle}$, where $\omega \in \mathbb{Z}_2^n$ and $\langle \omega \,|\, x \rangle$ is the $\mathbb{Z}_2^n$ inner product. Use $\langle f \,|\, g \rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{Z}_2^n} \overline{f(x)} g(x)$. Then the Fourier transforms of $f : \mathbb{Z}_2^n \to \mathbb{C}$ are:

$$\hat{f}(\omega) = \langle \omega \,|\, f \rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{Z}_2^n} (-1)^{\langle \omega \,|\, x \rangle} f(x) \quad \text{and} \quad f(x) = \sum_{\omega \in \mathbb{Z}_2^n} (-1)^{\langle \omega \,|\, x \rangle} \hat{f}(\omega).$$

The matrix of **DFT** decomposes as the n-fold Hadamard product:

$$\mathbf{DFT} = \mathbf{DFT}_2^{\otimes n} = \mathbf{DFT}_2 \otimes \mathbf{DFT}_2 \otimes \cdots \otimes \mathbf{DFT}_2$$

of the matrix $\mathbf{DFT}_2$ from the previous example.

5. Quantum Gates

5.1. **Pauli Matrices.** Suppose we wish to find universal gate sets that models how 2×2 unitary operators act on a single qubit. Moving to the Bloch sphere representation, each unitary operator on $\mathbb{C}^2$ is going to induce an operator on the sphere. One can convince oneself that the Bloch sphere operators corresponding to any *universal* quantum gate must necessarily include all 3d rotations, i.e. the special orthogonal group $SO(3)$.

Remark 5.1 (Unitaries give all rotations of the Bloch sphere). The unitary group $\mathrm{U}(n)$ decomposes as $\mathrm{SU}(n) \rtimes \mathrm{U}(1)$, where $\mathrm{SU}(n)$ is the special unitary group. When $n = 2$, the group $\mathrm{SU}(2)$ is isomorphic to the group of **versors**: the quaternions with unit norm. An isomorphism is given by:

$$q = u\mathbf{1} + a\mathbf{i} + b\mathbf{j} + c\mathbf{k} \mapsto U = \begin{bmatrix} \alpha & \beta \\ -\bar{\beta} & \bar{\alpha} \end{bmatrix} \quad \text{where} \quad \alpha = u + ai, \beta = b + ci.$$

Note that $a^2 + b^2 + c^2 + d^2 = |\alpha|^2 + |\beta|^2 = 1$. Identifying the span of $\mathbf{i}, \mathbf{j}, \mathbf{k}$ with $\mathbb{R}^3$, the conjugation map $v \in \mathbb{R}^3 \mapsto qvq^{-1} \in \mathbb{R}^3$ defines a rotation of $\mathbb{R}^3$ around the vector (a, b, c) by an angle of 2θ with $\cos\theta = u$ and $|\sin\theta| = \|(a, b, c)\|$. From this description, one might see that the map $v \in \mathbb{R}^3 \mapsto (-q)v(-q^{-1}) \in \mathbb{R}^3$ defines the same rotation. This gives a 2-1 surjective homomorphism $\mathrm{SU}(2) \twoheadrightarrow SO(3)$.

Example 5.2. We begin by first considering only rotations by π radians along each coordinate axis of the Bloch sphere. As special orthogonal transformations of $\mathbb{R}^3$, these are given by:

$$R_x(\pi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad R_y(\pi) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad R_z(\pi) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Each rotation fixes one of the X, Y, Z-bases of Definition 3.4 and swaps the other two. For example, the rotation along the z-axis swaps the X and Y bases: $|+\rangle \leftrightarrow |-\rangle$ and $|R\rangle \leftrightarrow |L\rangle$.

Definition 5.3 (Pauli matrices). The unitary operators corresponding to $R_x(\pi)$, $R_y(\pi)$, and $R_z(\pi)$ are:

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

Unsurprisingly, σ_x is called the **quantum NOT gate** and denoted by NOT. Next, σ_z is called the **phase flip gate**. Finally, $\sigma_y = i\sigma_x\sigma_z$ can be seen as a composition of a phase flip and NOT.

Example 5.4. To see how σ_x and $R_x(\pi)$ relate, note that $R_x(\pi)$ fixes $|+\rangle, |-\rangle$ and swaps the Y and Z bases. The $|0\rangle \leftrightarrow |1\rangle$ swap is clear from the matrix representation, i.e. $\sigma_x(1,0) = (0,1)$ and $\sigma_x(0,1) = (1,0)$. To see the Y basis swap:

$$\sigma_x|R\rangle = \sigma_x \begin{bmatrix} 1/\sqrt{2} \\ i/\sqrt{2} \end{bmatrix} = \begin{bmatrix} i/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}.$$

Multiply by $-i$ to eliminate the phase from the first factor to get $|L\rangle = (1/\sqrt{2}, -i/\sqrt{2})$. Similar calculations apply to σ_y and σ_z.

5.2. **Hadamard Gate.**

Definition 5.5. The **Hadamard gate** $H : \mathbb{C}^2 \to \mathbb{C}^2$ is defined as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \mapsto \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Remark 5.6. By Example 4.24, the Hadamard gate is the unitary DFT on $\mathbb{Z}_2$. It is also the quantum analog of a coin flip, in the sense that:

$$H(1,0)^\top = H(0,1)^\top = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \pm\frac{1}{\sqrt{2}} \end{bmatrix} \overset{|-|^2}{\rightsquigarrow} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}.$$

Starting on $|0\rangle$ or $|1\rangle$, applying H, and measuring returns $|0\rangle$ or $|1\rangle$ with equal probability. The Hadamard gate is incredibly important since it allows us to "create entanglement".

5.3. **Controlled NOT.** The controlled NOT gate acts on two qubits $\mathbb{C}^2 \otimes \mathbb{C}^2 \cong \mathbb{C}^4$. We introduce it in 2×2 block form, to emphasize their action on $\mathbb{C}^2 \otimes \mathbb{C}^2$.

Definition 5.7. The **controlled NOT** gate is given by:

$$\text{CNOT} = \begin{bmatrix} I & 0 \\ 0 & \text{NOT} \end{bmatrix}.$$

Example 5.8 (2-qubit (dis)entanglement). $H\otimes 1$ acts on two quibts. Its 4×4 matrix is obtained via the Kronecker product of H and I_2:

$$H\otimes 1 = \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ I & -I \end{bmatrix}.$$

Fix a basis for $|0\rangle, |1\rangle$ for $\mathbb{C}^2$. From these, obtain four 2-qubits basis vectors for $\mathbb{C}^2\otimes\mathbb{C}^2$:

$$|00\rangle = |0\rangle\otimes|0\rangle, \quad |01\rangle = |0\rangle\otimes|1\rangle, \quad |10\rangle = |1\rangle\otimes|0\rangle, \quad |11\rangle = |1\rangle\otimes|1\rangle.$$

Applying $H\otimes 1$ followed by a CNOT allows us to change between the computational basis and the Bell basis.

$$\begin{aligned} |00\rangle \mapsto |\Phi^+\rangle &= \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right), \\ |01\rangle \mapsto |\Phi^-\rangle &= \frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right), \\ |10\rangle \mapsto |\Psi^+\rangle &= \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right), \\ |11\rangle \mapsto |\Psi^-\rangle &= \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right). \end{aligned}$$

Since both operations are involutive, we can get back the computational basis by applying CNOT then $H\otimes 1$ to the Bell basis.

5.4. **Toffoli Gate.** Recall that in the world of classical computation, the Toffoli gate:

$$\text{Tof} : \mathbb{F}_2^3 \to \mathbb{F}_2^3 \quad (x, y, z) \mapsto \big(x, y, z \oplus (x \wedge y)\big)$$

alone formed a universal gate set (we could use it to build any Boolean function via circuits). The Toffoli gate is more or less the 3-bit analog for CNOT, since the latter can be expressed

as $(x, y) \mapsto (x, x \oplus y)$. Alternatively, compare matrix representations:

$$\text{CNOT} = \begin{bmatrix} I & \\ & \text{NOT} \end{bmatrix} \qquad \text{Tof} = \begin{bmatrix} I & & & \\ & I & & \\ & & I & \\ & & & \text{NOT} \end{bmatrix}.$$

So one way to think of these operators is as the quantum analogs of basic building blocks of classical computation, which will in turn become basic building blocks for quantum computation once we start implementing stuff in quantum computers. ‘

6. The Quantum Difference

We describe properties and protocols that make quantum computation different from classical computation.

6.1. **Decoherence.** If you start with classical information, use it to prepare a quantum system, and then immediately measure your system, you should end up with your same classical information. This is proven by Born's rule:

$$\langle i \,|p_j|\, i\rangle = \langle i \,|\, j\rangle\langle j \,|\, i\rangle = |\langle i \,|\, j\rangle|^2,$$

i.e. the probability of outcome j after preparing state i is $|\langle i \,|\, j\rangle|^2 = \delta_{ij}$.

Conversely, if you measure a quantum system starting in some mixed state $m = \Sigma_{ij} c_{ij} |i\rangle\langle j|$ to get a classical measurement, and then use that immediately to prepare a new quantum system, what do you get? The classical measurement will result in outcome $|i\rangle$ with probability $\text{Tr}(p_i m) = \langle i \,|m|\, i\rangle = \Sigma_{ij} c_{ij} \langle i \,|\, i\rangle\langle j \,|\, i\rangle = \Sigma_i c_{ii}$.

$$\langle i \,|m|\, i\rangle = \Sigma_i c_{ij} \langle i \,|\, i\rangle\langle j \,|\, i\rangle =$$

6.2. **Super Dense Coding.** (see Christos's note)

6.3. **Quantum Teleportation.** In this scenario, Alice has (1) a classical channel, (2) access to her half of a shared entangled 2-qubit $|\Phi^+\rangle$, and (3) a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ whose state

she wants to send to Bob. The total system then has 3-qubits:

$$\begin{aligned}|\psi\rangle\otimes|\Phi^+\rangle &= \Big(\alpha|0\rangle+\beta|1\rangle\Big)\otimes\frac{1}{\sqrt{2}}\Big(|00\rangle+|11\rangle\Big)\\ &= \frac{\alpha}{\sqrt{2}}|000\rangle+\frac{\alpha}{\sqrt{2}}|011\rangle+\frac{\beta}{\sqrt{2}}|100\rangle+\frac{\beta}{\sqrt{2}}|111\rangle.\end{aligned}$$

Alice performs a change of basis to the slots she can access (the first two), from the computational basis to the Bell basis (using the formulas above). For example, the first term becomes:

$$\frac{\alpha}{\sqrt{2}}|000\rangle=\frac{\alpha}{\sqrt{2}}|00\rangle\otimes|0\rangle=\frac{\alpha}{2}\left(|\Phi^+\rangle+|\Phi^-\rangle\right)\otimes|0\rangle.$$

Doing this for all four terms and simplifying gives:

$$\begin{aligned}|\psi\rangle\otimes|\Phi^+\rangle =&\frac{1}{2}\Big[|\Phi^+\rangle\otimes(\alpha|0\rangle+\beta|1\rangle)+|\Phi^-\rangle\otimes(\alpha|0\rangle-\beta|1\rangle)\\ &+|\Psi^+\rangle\otimes(\alpha|1\rangle+\beta|0\rangle)+|\Psi^-\rangle\otimes(\alpha|1\rangle-\beta|0\rangle)\Big]\end{aligned}$$

Notice we changed basis and switched from $\mathbb{C}^2\otimes(\mathbb{C}^2\otimes\mathbb{C}^2)\cong(\mathbb{C}^2\otimes\mathbb{C}^2)\otimes\mathbb{C}^2$, so nothing has changed numerically. However, now Bob's qubit (third slot) almost all look like $|\psi\rangle$.

The 2-qubit of $|\psi\rangle\otimes|\Phi^+\rangle$ that Alice can now access is $\frac{1}{2}|\Phi^+\rangle+\frac{1}{2}|\Phi^-\rangle+\frac{1}{2}|\Psi^+\rangle+\frac{1}{2}|\Psi^-\rangle$. When she takes a measurement of it, the 3-qubit collapses, and she will observe exactly one of the Bell states, each with uniform probability $|\frac{1}{2}|^2=\frac{1}{4}$. If Bob were to measure *his* qubit at this point, he would get the third term of the summand corresponding to the Bell state Alice saw (e.g. if Alice saw $|\Psi^-\rangle$, Bob would see $\alpha|1\rangle-\beta|0\rangle$).

So depending on which Bell state she observes, Alice's new goal is to find an operator that will transform Bob's qubit to the original state $|\psi\rangle=\alpha|0\rangle+\beta|1\rangle$.

(1) If Alice sees $|\Phi^+\rangle$, Bob's qubit is $|\psi\rangle$, so he should do nothing (identity transformation).

(2) If Alice sees $|\Phi^-\rangle$, Bob's qubit is $\alpha|0\rangle-\beta|1\rangle$, so he needs to negate $|1\rangle$.

$$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}\alpha\\ -\beta\end{bmatrix}=\begin{bmatrix}\alpha\\ \beta\end{bmatrix}.$$

(3) If Alice sees $|\Psi^+\rangle$, Bob's qubit is $\alpha|1\rangle + \beta|0\rangle$, so he needs to swap $|0\rangle$ and $|1\rangle$.

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \alpha \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

(4) If Alice sees $|\Psi^-\rangle$, Bob's qubit is $\alpha|1\rangle - \beta|0\rangle$, so he needs to negate $|0\rangle$ and then swap $|0\rangle$ and $|1\rangle$.

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -\beta \\ \alpha \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

If Bob knows which transformation to apply, he is guaranteed to observe $|\psi\rangle$ once he measures. Therefore, all Alice and Bob need to do is establish a bijection between the four matrices above and $\mathbb{F}_2^2 = \{00, 01, 10, 11\}$ before departing. If they do, then Alice can send 2-bits of information across the classical channel to convey which operator Bob should apply.

The entire process from start to finish thus looks like this:

(1) Alice and Bob fix a basis $|0\rangle, |1\rangle$ for their qubits and a bijection:

$$\{00, 01, 10, 00\} \quad \leftrightarrow \quad \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right\}.$$

(2) The entangled 2-qubit Bell state $|\Phi^+\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2$ is created.

(3) Alice and Bob move far away, each getting one qubit from $|\Phi^+\rangle$.

(4) Alice obtains an extra qubit $|\psi\rangle$ whose *state* she wishes to communicate to Bob.

(5) Alice takes the composite 3-qubit $|\psi\rangle \otimes |\Phi^+\rangle$ and changes basis for the first two slots, from the computational basis to the Bell basis.

(6) Alice measures her $|\psi\rangle$ qubit and her $|\Phi^+\rangle$ qubit.

(7) The entanglement collapses, and Alice observes one of 4 Bell states.

(8) Alice sends a 2-bit message to Bob across the classical channel.

(9) Bob receives the message and applies the corresponding operator to his qubit.

(10) Bob takes a measurement and observes $|\psi\rangle$.

In particular, notice that the 2-bit message may be intercepted, and the secret code (bijection) may be leaked, but neither compromises the state of $|\psi\rangle$.

7. Algorithms

7.1. **Grover's Search Algorithm.** Suppose you're given a classical machine $\delta_a : \mathbb{F}_2^n \to \mathbb{F}_2$ that outputs 1 when $x = a$ and 0 otherwise. Using a brute force search, we will have to run δ_a an average of $2^n/2$ times to find a.

Now suppose you're given a checking machine $f : \mathbb{F}_2^n \to \mathbb{F}_2$ that outputs 1 for $x = a$ and 0 otherwise. The goal of Grover's algorithm is to find $|a\rangle$ at less cost/runtime.

We will use an average over all observable states:

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{I \in \mathbb{F}_2^n} |I\rangle.$$

Notice that, even without knowing what $|a\rangle$ is, $|s\rangle$ is equidistant from all the standard n-qubit basis vectors, so $\langle s \,|\, a\rangle = \frac{1}{\sqrt{2^n}}$. We also define:

$$|s'\rangle = \frac{1}{\sqrt{2^n - 1}} \sum_{I \neq a} |I\rangle,$$

because (geometrically) the operators we will perform will only move around the plane spanned by $|a\rangle$ and $|s\rangle$, so it will be useful to have an orthonormal basis $|s'\rangle, |a\rangle$ for $\text{span}(|a\rangle, |s\rangle)$.

Here are the unitary operators we use in Grover's algorithm:

- The Hadamard gate H that maps $|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
- The unitary operator U_a mapping $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$, i.e. reflection across the hyperplane orthogonal to $|a\rangle$. In the plane $\text{span}(|a\rangle, |s\rangle)$, U_a corresponds to reflection across the hyperplane orthogonal to $|s'\rangle$. Notice that U_a is implement through our access to the checker f, so we are able to implement it without knowing what a is.
- The reflection across the hyperplane orthogonal to $|s\rangle$.

7.2. **Simon's Algorithm.** We are given a black box function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ with a secret point $a \in \mathbb{F}_2^n$ such that $f(x) = f(y)$ iff $x = y \oplus a$. Our goal is to find a, i.e. find the period of the function f. Notice that f is 1-1 when $a = 0$ and 2-1 otherwise. We assume $a \neq 0$ and set the total number of inputs to be $N = 2^n$.

To solve this problem classically, we would need to run f until we find two inputs x_1, x_2 such that $f(x_1) = f(x_2)$. In that case, $x_1 = x_2 \oplus a$, so XOR-ing both sides by x_2 gives

$a = x_1 \oplus x_2$. In the worst case, we will have to run f a total of $N/2+1$ times to get a repeat or conclude f is injective. In general, the expected value is $\sim \sqrt{N}$.

For the quantum case, given f as a black box, we can implement a unitary operator U_f acting on two n-qubits as $|x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |y \oplus f(x)\rangle$. In particular, running U_f with $|y\rangle = |0\rangle$ recovers f. Simon's algorithm will perform the following operations:

$$(H^{\otimes n} \otimes 1) \circ U_f \circ (H^{\otimes n} \otimes 1)$$

starting with $|\varphi_0\rangle = |0\rangle \otimes |0\rangle$.

After the first step, we obtain our usual uniform superposition on the first n-qubit:

$$|\varphi_1\rangle = (H^{\otimes n} \otimes 1)(|0\rangle \otimes |0\rangle) = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{F}_2^n} |x\rangle \otimes |0\rangle.$$

After the second step, we spread $f(x)$ to our second n-qubit:

$$|\varphi_2\rangle = U_f(|\varphi_1\rangle) = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{F}_2^n} |x\rangle \otimes |f(x)\rangle$$

Finally, the third step gives:

$$\begin{aligned} |\varphi_3\rangle &= \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{F}_2^n} (H^{\otimes n} \otimes 1)(|x\rangle \otimes |f(x)\rangle) \\ &= \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{F}_2^n} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{F}_2^n} (-1)^{\langle x \,|\, y \rangle} |y\rangle \otimes |f(x)\rangle \\ &= \frac{1}{N} \sum_{x,y \in \mathbb{F}_2^n} (-1)^{\langle x \,|\, y \rangle} |y\rangle \otimes |f(x)\rangle. \end{aligned}$$

By assumption, $|y\rangle \otimes |f(x)\rangle = |y\rangle \otimes |f(x) \oplus a\rangle$, so each ket appears twice, i.e.

$$\begin{aligned} |\varphi_3\rangle &= \frac{1}{N} \sum_{y \in \mathbb{F}_2^n} \sum_{\text{half of } x \in \mathbb{F}_2^n} \left((-1)^{\langle x \,|\, y \rangle} + (-1)^{\langle x \oplus a \,|\, y \rangle} \right) |y\rangle \otimes |f(x)\rangle \\ &= \frac{1}{N} \sum_{x,y \in \mathbb{F}_2^n} \frac{1}{2} \left((-1)^{\langle x \,|\, y \rangle} + (-1)^{\langle x \oplus a \,|\, y \rangle} \right) |y\rangle \otimes |f(x)\rangle, \end{aligned}$$

where the last sum has all distinct kets. From basic properties of the inner product on $\mathbb{F}_2^n$, we get that $\langle x \oplus a \,|\, y \rangle = \langle x \,|\, y \rangle \oplus \langle c \,|\, y \rangle$ and that $(-1)^{\langle x \,|\, y \rangle \oplus \langle c \,|\, y \rangle} = (-1)^{\langle x \,|\, y \rangle} (-1)^{\langle c \,|\, y \rangle}$. Thus,

each term's coefficient looks like:

$$\frac{1}{2}\left((-1)^{\langle x\,|\,y\rangle} + (-1)^{\langle x\,|\,y\rangle}(-1)^{\langle a\,|\,y\rangle}\right)$$
$$= \frac{1}{2}(-1)^{\langle x\,|\,y\rangle}\left(1 + (-1)^{\langle a\,|\,y\rangle}\right).$$

Thus the coefficient is determined by $\langle a\,|\,y\rangle \in \{0,1\}$. In particular, the linear functional $\langle a\,|\,-\rangle : \mathbb{F}_2^n \to \mathbb{F}_2$ is surjective with kernel of size $N/2$. If $\langle y\,|\,a\rangle = 1$ the coefficient is zero, and if $\langle y\,|\,a\rangle = 0$ it is ± 1. Hence, upon measuring the first n-qubit, we will only get one of the $N/2$ binary strings y such that $\langle a\,|\,y\rangle = 0$.

How many times will we have to run this process to find $|a\rangle$? Let $S = \operatorname{span}\{a\}$ in $\mathbb{F}_2^n$. Then each run returns a binary strings $y \in S^\perp$. We need to find enough y's to span the $n-1$ dimensional subpsace $S^\perp$. Computationally, each y gives a linear equation $\langle a\,|\,y\rangle = 0$, and we need $n-1$ independent equations to fully solve a system for a.

Suppose that after some runs, you've measured y_i's with $\operatorname{span}\{y_i\} = r < n-1$. The probability that your next measurement is new is:

$$1 - \frac{|\operatorname{span}\{y_i\}|}{S^\perp} = 1 - \frac{2^r}{2^{n-1}} = 1 - 2^{r+1-n}.$$

Therefore, each new measurement on average increases the dimension of the span by the reciprocal of the number above. To fully obtain $S^\perp$ we need r to range from 0 to $n-2$. Summing all these give the total expected number of runs needed:

$$\sum_{k=1}^{n-1} \frac{1}{1-2^{k-n}} = \sum_{k=1}^{n-1} 1 + \frac{2^{-k}}{1-2^{-k}} = (n-1) + \sum_{k=1}^{n-1} \frac{2^{-k}}{1-2^{-k}}.$$

The second sum is $\sum_{k=1}^{\infty} \frac{1}{2^k-1}$, which converges to something between 1 and 2. Thus, in general, we need around $n+1$ runs to find $|a\rangle$.

7.3. **Miller-Rabin algorithm.**

Theorem 7.1 (Fermat's Little Theorem). For prime p, $(\mathbb{Z}/p\mathbb{Z})^*$ is cyclic order $p-1$. For all $a \neq 0$ we thus have $a^{p-1} \equiv 1 \mod p$.

Definition 7.2. For a group G, let ϕ^m denote the m-th power map $x \mapsto x^m$. Denote $G^{(m)} = \operatorname{im}\phi_m$ and $G_{(m)} = \ker\phi_m$.

Note that the fibers of ϕ_m have the same cardinality, so if a is uniformly distributed in G, then so is a^m in G.

Theorem 7.3 (Naive Primality Test). Let $N \in \mathbb{N}$. Pick $1 \leq a \leq N-1$ uniformly.

(1) If $a^{N-1} \not\equiv 1 \mod N$ then N is composite. (end)

(2) If $a^{N-1} \equiv 1 \mod N$ then N is probably prime. Fails when $a \in (\mathbb{Z}/N\mathbb{Z})^*_{(N-1)}$.

If there's an $a^{N-1} \not\equiv 1 \mod N$ (i.e. when $((\mathbb{Z}/N\mathbb{Z})^*)^{(N-1)}$ is nontrivial), then this test detects compositeness with probability $> \frac{1}{2}$. The test fails for composite numbers N such that $a^{N-1} \equiv 1 \mod N$ for all $a \neq 0$.

Proof. Assume $N > 3$ is composite and that there's some a for which $a^{N-1} \not\equiv 1 \mod N$. The second assumption means that $(\mathbb{Z}/N\mathbb{Z})^*_{(N-1)} \subsetneq (\mathbb{Z}/N\mathbb{Z})^*$. We will show that $(\mathbb{Z}/N\mathbb{Z})$ is at least double the size, so that the chance of error will be $\frac{|(\mathbb{Z}/N\mathbb{Z})^*_{(N-1)}|}{|(\mathbb{Z}/N\mathbb{Z})^*|} < \frac{1}{2}$. This follows by simply looking at the quotient $(\mathbb{Z}/N\mathbb{Z})^*/(\mathbb{Z}/N\mathbb{Z})^*_{(N-1)}$ which has order ≥ 2. □

Exercise 7.4 (3.4.6). If $N = p^c$ with $c > 1$ is an odd prime power, then $((\mathbb{Z}/N\mathbb{Z})^*)^{(N-1)}$ is nontrivial, i.e. $\exists a$ with $a^{N-1} \not\equiv 1 \mod N$ and step 1 succeeds with probability $> \frac{1}{2}$.

Proof. Set $a = p^c + 1 - p^{c-1}$. Set $x = p^{c-1}$. Then:

$$\begin{aligned}
a^{N-1} &= (p^c + 1 - p^{c-1})^{p^c - 1} \\
&\equiv (1 - p^{c-1})^{p^c-1} \\
&= (1-x)^{p^c-1} \\
&= \sum_{k=0}^{p^c-1} \binom{p^c-1}{k} (-x)^k \\
&\equiv 1 - (p^c - 1)x + (\text{terms divisible by } p) \\
&= 1 + x \\
&= 1 + p^{c-1}.
\end{aligned}$$

If $a^{N-1} \equiv 1$ then $1 + p^{c-1} \equiv 1$, i.e. $p^{c-1} \equiv 0 \mod p^c$, which can't happen. □

We use the exercise above, combined with Sunzi's remainder theorem, to create a terminating algorithm with probability of success $> \frac{1}{2}$.

Theorem 7.5 (Miller-Rabin Algorithm). Let $N \in \mathbb{N}$ be odd. Pick $1 \leq a \leq N-1$ uniformly.

(1) If $a^{N-1} \not\equiv 1 \mod N$ then N is composite. (end)

(2) If $a^{N-1} \equiv 1 \mod N$ then:

   (a) Find largest power $2^k \mid N-1$. Write $N-1 = 2^k \ell$.

   (b) If $a^\ell \not\equiv 1 \mod N$ then N is composite.

   (c) If $a^{2^j \ell} \not\equiv -1 \mod N$ then N is composite.

(3) Else, report N is prime (high probability of success).

This reports the correct answer on any input with probability $> \frac{1}{2}$.

Proof. To explain steps 2b and 2c, note:

$$\begin{aligned} a^{N-1} - 1 &= a^{2^k \ell} - 1 \\ &= (a^{2^{k-1}\ell})^2 - 1 \\ &= (a^{2^{k-1}\ell} - 1)(a^{2^{k-1}\ell} + 1) \\ &= (a^{2^{k-2}\ell} - 1)(a^{2^{k-2}\ell} + 1)(a^{2^{k-1}\ell} + 1) \\ &\vdots \\ &= (a^\ell - 1)(a^\ell + 1)(a^{2\ell} + 1)(a^{2^2\ell} + 1) \cdots (a^{2^{k-1}\ell} + 1). \end{aligned}$$

If N is prime, then $\mathbb{Z}/N\mathbb{Z}$ is an integral domain, so the only way for $a^{N-1} - 1 \equiv 0 \mod N$ is for one of the factors to be zero. Taking the contrapositive, if one of the factors is nonzero, then N is composite. We check the probability of failure is $< \frac{1}{2}$. To get a failure, assume N is an odd composite, and that we picked an a uniformly that satisfied $a^{N-1} \equiv 1 \mod N$, i.e. a was picked uniformly from $(\mathbb{Z}/N\mathbb{Z})^*$.

Decompose $N = uv$ with u, v odd and coprime. By Sunzi's remainder theorem:

$$((\mathbb{Z}/N\mathbb{Z})^*)^{(N-1)} \cong ((\mathbb{Z}/u\mathbb{Z})^*)^{(N-1)} \times ((\mathbb{Z}/v\mathbb{Z})^*)^{(N-1)}.$$

If either factor is nontrivial, then step 1 will succeed with probability $> \frac{1}{2}$, so assume they're both trivial.

For step 2, we consider powers $a^{2^j \ell}$ and want to find a j with probability $> \frac{1}{2}$ such that $a^{2^j \ell} \not\equiv \pm 1 \mod N$ and $a^{2^{j+1}\ell} \equiv 1 \mod N$. $\square$

Theorem 7.6. Primality testing is in **P**.

The core of the proof follows from:

Lemma 7.7. N is prime iff $(x+a)^N \equiv x^N + a^N \mod N$ in $(\mathbb{Z}/N\mathbb{Z})[x]$.

Proof. Forwards direction is just binomial expansion. For the converse, prove the contrapositive. If $N = ap^m$ is composite, then the p-th coefficient of the expansion is:

$$\binom{N}{p} = \frac{N!}{p!(N-p)!} = \frac{N}{p}\frac{(N-1)!}{(p-1)!(N-p)!} = \frac{N}{p}K.$$

Note $N/p \in \mathbb{Z}$ and K is not divisible by p, so this term is nonzero in the polynomial ring. □

7.4. **Shor's Algorithm (Classical).** Schor's algorithm involves a quantum part and a classical part. The quantum part takes a random $a \in (\mathbb{Z}/N\mathbb{Z})^*$ and tries to find its order, i.e. finds the smallest integer r such that $a^r - 1 = mN$. Suppose we have $a \in (\mathbb{Z}/N\mathbb{Z})^*$ and its order r.

The classical part uses this information to find a factor of N with probability $> 1/4$. Thus, the probability of success after k runs is $> 1 - 0.75^k$, i.e. $> 25\%$, after one run, $> 43.25\%$ after two runs, $> 57.81\%$ after three runs, and $> 68.36\%$ after four runs.

The classical part succeeds when r is even and $a^{r/2} \not\equiv -1 \mod N$. Both of these have probability of occurring greater than $1/2$, so we get success with probability $> 1/4$. When these conditions are met, we may write $a^r - 1 = (a^{r/2}-1)(a^{r/2}+1)$ with neither factor being zero in $\mathbb{Z}/N\mathbb{Z}$. Applying the division algorithm to N and $a^{r/2} - 1$ finds a factor of N.

Exercise 7.8. Let G be abelian of even order. Show the number of elements of even order is greater than or equal to the number of elements of odd order.

Proof. Base Case: Suppose G is cyclic, i.e. $G = \mathbb{Z}/2m\mathbb{Z}$ for some $m \in \mathbb{Z}$. Let $x \in G$ have odd order k, so $2m$ divides kx. Consider the order of $x+1$. If the order k' is also odd then $2m$ divides $k'(x+1)$, so $2m$ divides $(k+k')x + k'$. This makes the latter an odd number divisible by $2m$, which is a contradiction. Thus $x+1$ had even order.

General: For general G, write $G \cong \Pi_i \mathbb{Z}/N_i\mathbb{Z}$ as a product of cyclic groups, with the first entry having even order. If $(x_i) \in G$ has odd order, each x_i has odd order. But then by the base case, $(x_1 + 1, x_2, x_3, \dots)$ has even order. □

Email address: fllfernando95@gmail.com

URL: https://fernando-liu-lopez.github.io/fernandoliulopez.github.io/