CS 4510

Quantum Computing CheatSheet

Name: Abrahim Ladha

1.1 Physics Notation

We describe a quantum state as a vector of the form $|\psi\rangle \in V$. In a complex valued vector space V. For any number of vectors in this state, we may superposition them as a linear combination of complex coefficients. Given some basis of our vector space, we may write any $|\psi\rangle = \sum_{i=1}^{n} c_i |e_i\rangle$. The dimension of our vector space in general is infinite (making it a Hillbert space). Any quantum state can be described by some ket[1] in our Hillbert space.

For each $|\psi\rangle \in V$, there exists $\langle\psi| \in V^{*2}$ such that $\langle\psi| = |\psi\rangle^{\dagger}$.[3]

The inner product is written naturally as $\langle\phi|\psi\rangle = |\phi\rangle^{\dagger} |\psi\rangle$. the sum of the pairwise components, and is a complex number. Written out:

$$\langle\phi|\psi\rangle = \sum_{i=1}^{n} \phi_i \psi_i \tag{1.1}$$

There is also a less intuitive outer product defined as a matrix.

$$|\phi\rangle \langle\psi| = \begin{bmatrix} \phi_1\psi_1 & \phi_1\psi_2 & \dots & \phi_1\psi_n \\ \phi_2\psi_1 & \dots & \dots & \dots \\ \dots & \dots & & \\ \phi_1\psi_n & \dots & \dots & \phi_n\psi_n \end{bmatrix} \tag{1.2}$$

This is an operator less naturally. Suppose $A = |\phi\rangle \langle\psi|$, then $A|x\rangle = |\phi\rangle \langle\psi|x\rangle = c|\phi\rangle$. While The inner product is a complex value, since the outer product with a vector is another vector, it must be an operator.

1.2 Quantum Computer

1.2.1 Qubits

A *classical computer* is simply a Turing machine, or some model of equal power. There is some infinite tape where each cell may contain a 0 or 1. Instead of bits, a *quantum computer* has qubits, which can be a 0, 1 or any *superposition* of 0 or 1. More formally, instead of 0 or 1, we have a zero or one vector, denoted as $|0\rangle$ and $|1\rangle$.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad\qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{1.3}$$

[1] $|\cdot\rangle$ is called a ket, and $\langle\cdot|$ is called a bra. So a $\langle\cdot|\cdot\rangle$ is called a braket. Like bracket. Funny right?

[2] spoken as: "bra in the dual space"

[3] The $\dagger$ operator is the hermitian conjugate. It is the complex conjugate, and the transpose. The complex conjuage just replaces i with $-i$ component wise. If $|\psi\rangle$ is a column vector, then its transpose, $\langle\psi|$ is a row vector

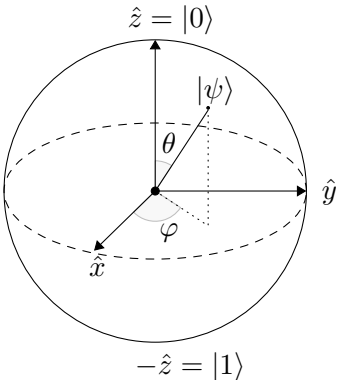


Figure 1.1: The Bloch sphere.

Our superposition is $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$. If you ever try to *measure* a qubit, you will only see that you will get $|0\rangle$ or $|1\rangle$. If you try to measure it, you will only receive $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$.[4] Given this construction, we can then represent a qubit in the form

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\,|1\rangle \tag{1.4}$$

Reparametrizing our qubit $|\psi\rangle$ in terms of θ, φ as angles, we get the analogy of a qubit being in a state as

1.2.2 Multiple Qubits

A single qubit is rather useless. The power of a quantum computer involves computing on multiple qubits simultaneously. The tensor[5] product of two quantum states is as defined:

$$|\phi\rangle \otimes |\psi\rangle = \begin{bmatrix} \phi_1 \begin{bmatrix} \psi_1 \\ \dots \\ \psi_n \end{bmatrix} \\ \dots \\ \phi_n \begin{bmatrix} \psi_1 \\ \dots \\ \psi_n \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \phi_1\psi_1 \\ \phi_1\psi_2 \\ \dots \\ \phi_n\psi_n \end{bmatrix} \tag{1.5}$$

This is intuitively similar like a cartesian product. It is common for the symbol to be dropped, and simply be written as $|\phi\rangle\,|\psi\rangle$.

A quantum state which cannot be written as a tensor product of qubit states is called "entangled". Without getting too much into the physics, two entangled qubits have their destinies tied, until unentangled. You cannot operate on one qubit without something changing on the other.

[4] You might think we have four degrees of freedom here with α, β being complex numbers with two real parts each, but we only have three because the constraint $|\alpha|^2 + |\beta|^2 = 1$. Also recall, these are not absolute values, but norms. The norm of a complex number is essentially, its length as a vector in $\mathbb{R}^2$. You may compute it as $|\alpha| = \sqrt{\alpha^*\alpha}$. For example $|2 + 3i| = \sqrt{(2 - 3i)(2 + 3i)} = \sqrt{4 + 9} = \sqrt{13}$.

[5] Some people call this kronecker product

1.2.3 Quantum Gates

A quantum gate is simply a single, or multi qubit unitary operator. Four common gates are as follows:

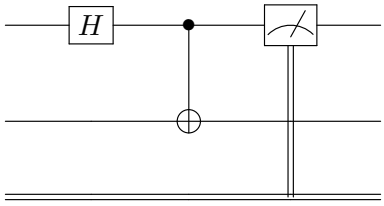
$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{1.6}$$

Under the geometric idea, these represent rotations of 180 degrees upon different axii. Another common gate is the Hadamard transform. It is defined recursively as $H_m = H_1 \otimes H_{m-1}$, with $H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and $H_1 = \begin{bmatrix} 1 \end{bmatrix}$. Another common gate is called $CNOT$. It operates on two qubits. If both qubits are $|0\rangle, |1\rangle$, then it behaves classically like XOR. We define it as

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.7}$$

1.2.4 Quantum Circuits

A Quantum circuit looks like this.



This is a circuit of two qubits and one classical bit. We initialise in state $|00\rangle$. Then a Hadamard gate is applied to the first qubit, giving us $H|0\rangle \otimes |0\rangle$. A CNOT gate is then applied, giving us $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. This is called a Bell state, and is maximally entangled. Finally, the last gate is a measurement operation. It measures the first qubit, and writes it to the classical wire. A circuit to implement something like Shor's algorithm may be on the scale of thousand of qubits and tens of thousands of gates. From this, it is not immediately obvious that quantum circuits are Turing complete, even in a non-uniform sense. That is beyond the scope of this document.

1.2.5 No Cloning Theorem

It is impossible to copy or clone a qubit. Intuitively, you may think that the cloning operation requires the ability to take nondestructive measurement. We can prove there is no arbitrary operations that can clone a qubit.

Assume to the contrary there is some unitary transform that can clone qubits. That is, given on input $|\psi\rangle|0\rangle$ it outputs $|\psi\rangle|\psi\rangle$. Written as operators, we have $|\psi\rangle|\psi\rangle = U|\psi\rangle|0\rangle$. Since this unitary transform works for all qubits, we have $|\varphi\rangle|\varphi\rangle = U|\varphi\rangle|0\rangle$, for any other qubit $|\varphi\rangle$. Lets inner product these two statements:

$$\langle\psi|\langle\psi|U^\dagger U|\varphi\rangle|\varphi\rangle = \langle 0|\langle\psi|\varphi\rangle|0\rangle \tag{1.8}$$

$$\langle\psi|\varphi\rangle^2 = \langle\psi|\varphi\rangle \tag{1.9}$$

This implies that $\langle\psi|\varphi\rangle$ is 0 or 1, which is not true to the assumption that $|\psi\rangle, |\varphi\rangle$ were any arbitrary states.

1.3 Shors Algorithm

1.3.1 Period finding

The period of a function f is some r such that for all x $f(x+r) = f(x)$. How can this be used for integer factorization? If $f(x) = a^x \pmod N$ then

$$f(x+r) = f(x) \tag{1.10}$$
$$a^{x+r} \equiv a^x \pmod N \tag{1.11}$$
$$a^r \equiv 1 \pmod N \tag{1.12}$$

Then for $a \in \mathbb{Z}_N$, We know $r|N$.

1.3.2 The classical portion

Shor's algorithm has two parts. A quantum portion to find the period in polynomial time, and some classical processing. For now, suppose that the quantum period finding portion is a polynomial time oracle. Suppose that we wish to factor $N = pq$. We can do processing to eliminate smaller factors before hand to eliminate the chance that we run Shor's algorithm to retrieve a factor we could have done classically anyway. For any composite N, it must contain a factor less than $\sqrt{N}$ so for p, q equal bitlength.[6], this is our ideal worst case. The algorithm is as follows:

1. $a \xleftarrow{\$} \mathbb{Z}_N$
2. Check if a, N are co-prime[7], If they are not then halt and return a
3. Query our oracle for $r \leftarrow QC(a, N)$
4. if r is odd, restart with a different a
5. if $a^{r/2} \equiv -1 \pmod N$ restart with a different a
6. Return one of $\gcd(a^{r/2} - 1, N)$ or $\gcd(a^{r/2} + 1, N)$

Why is this correct? Given that $a^r \equiv 1 \pmod N$, then $(a^{r/2} - 1)(a^{r/2} + 1) \equiv a^r - 1 \equiv 0 \pmod N$.

1.3.3 Quantum Portion

Beyond the scope of this document, but there is information online.

1.4 Violated Hardness Assumptions

As demonstrated, we have a quantum algorithm for integer factorization in polynomial time.

[6] I can factor a 256 bit number into its two factors, both around 127 bits in four minutes on my laptop. If $N = pq$ and $p >> q$, then $p >> \sqrt{N}$ and $q << \sqrt{N}$, so when we are trying to find any factor, we will find q much faster. For PGP, the weakest choice of an RSA modulus was 512 bits.

[7] Two numbers are a, b co-prime if they share no common factors. This can be tested by checking if $\gcd(a, b) = 1$. This is computed via the euclidean algorithm, which runs in polynomial time. If a, N are not co-prime, then $\gcd(a, N)$ is an factor of N not equal to 1.

1.4.1 RSA Example

The RSA problem is given $N, e, M^e \pmod N$, can you efficiently[8] compute M. You can break the RSA problem by breaking discrete log, but you can shortcut and break the RSA cryptosystem by breaking integer factorization. Given $N = pq, M^e \pmod N$, you can factor N efficiently to compute $\phi(N) = (p-1)(q-1)$, knowing this, and $ed \equiv 1 \pmod{\phi(N)}$, you can obtain d which lets you solve for $M = (M^e)^d$.

1.4.2 Discrete Log

Recall the discrete log problem. Given $y = g^x \pmod N$, solve for $x = DLOG_g(y)$ The current state of the art classical algorithm for solving the DLP is the Pollig Hellman algorithm and it runs in $O(\sqrt{N})$ in the worst case. Solving DLP implies you can factor integers in polynomial time, but the reverse remains an open question. Shor's algorithm can be modified to immediately break discrete log as follows:

1. Given $y \equiv g^x \pmod N, g, N$, the task is to solve for x. Suppose N is prime[9] and g is a generator
2. Construct the bivariate, periodic function $f(x_1, x_2) = g^{x_1} y^{x_2}$
3. Obtain the pair $(r_1, r_2) \leftarrow QC(g, y, N)$ from our quantum oracle such that $f(x_1, x_2) = f(x_1 + r_1, x_2 + r_2)$
4. return $x = -\frac{r_1}{r_2} \pmod N$

For N composite, this can be extended without knowing the factorization of N, which some messier cases. Why is this correct? $g_1^x y_2^x = g^{x_1+r_1} y^{x_2+r_2} \iff g^{r_1} y^{r_2} \equiv 1 \pmod N$, but then $g^{r_1} y^{r_2} \equiv g^{r_1} g^{xr_2} \equiv g^{r_1+xr_2} \equiv 1 \pmod N$. Since g is a generator, then $r_1 + xr_2 \equiv 0 \pmod N$. Solving for x we get $x \equiv -\frac{r_1}{r_2} \pmod N$.

1.5 Problems

1. For any qubit $|\psi\rangle$, What is $\langle\psi|\psi\rangle$?
2. Compute $|1\rangle\langle 1| + |0\rangle\langle 0|$ as a matrix.
3. Why is it important that quantum operators be unitary?
4. For an n qubit system, what is the length of the vector representing a quantum state of this system?
5. Give 3 distinct qubits which are uniform. By uniform, I mean upon measurement, there is equal chance to measure $|0\rangle$ or $|1\rangle$.
6. Without looking it up, what can you infer about the best running time for classical period finding? Consider Shor's algorithm as a reduction.

If you are interested in this stuff, there are some courses you can take. CS4782 Quantum Information and Quantum Computing, CS8803 (This is actually programming QC), ECE8863 Quantum Devices, a hardware class.

[8] "efficiently" means different things depending on what flavor of computer scientist you ask. If you ask a cryptographer, they mean in polytime.

[9] It works if N is not prime just as well, but requires the chinese remainder theorem, and repeated applications.