CSE439/501 Week 2: Linear Algebra, Unitary Operations, and Qubits

First, one item added to review of last lecture:

- $f(n) = O(g(n))$ means there is a constant $C > 0$ such that for all n past a certain point n_0, $f(n) \leq C \cdot g(n)$.
- $f(n) = \Theta(g(n))$ means there are constants c and C such that beyond some point n_0, $f(n)$ stays in the range $c \cdot g(n)$ to $C \cdot g(n)$.
- $f(n) \sim g(n)$ is stronger---it says that the limit $f(n)/g(n)$ exists and equals 1.

An example of the last is $f(n) = 3n^2$ versus $g(n) = 3n^2 + 100n + 500$.

The last bullet is the same as $\lim_{n \to \infty} \frac{g(n)}{f(n)} = 1$. In the example:

$$\frac{g(n)}{f(n)} = \frac{3n^2 + 100n + 500}{3n^2} = 1 + \frac{100n}{3n^2} + \frac{500}{3n^2} = 1 + \frac{100}{3n} + \frac{500}{3n^2}$$

The limit is 1 because the limits of the latter two fractions are 0.

Tensor Products

In a calculus or linear algebra course you have likely encountered the spaces $\mathbb{R}^2$ of points (a, b) in the plane and $\mathbb{R}^3$ of points (x, y, z) or (x_1, x_2, x_3) in 3-dimensional space. Then $\mathbb{R}^n$ means n-dimensional real space, whether you called it a vector space or not. Maybe you also covered the complex vector spaces $\mathbb{C}^n$ or specialized to vectors of rational numbers---which make the vector space $\mathbb{Q}^n$. When we care more about the "space" aspect than the particular kind of numbers allowed, we use the umbrella term "Hilbert space" after the mathematician David Hilbert. That term is often employed by physicists not only to avoid having to specify the dimension n but also to allow it to be infinite. We, however, will stay in finite dimensional spaces and care a lot about what the dimension is.

The usual rule for the *product* of two vector spaces is to add the dimensions. Thus a member of $\mathbb{R}^2 \times \mathbb{R}^3$, which formally is an ordered pair like $((a, b), (x, y, z))$, is considered the same as the 5-tuple (a, b, x, y, z), which we could re-label as $(x_1, x_2, x_3, x_4, x_5)$. So

$$\mathbb{R}^2 \times \mathbb{R}^3 = \mathbb{R}^5.$$

The **tensor product**, however, multiplies the dimensions. When defined between our vectors (a, b) and (x, y, z), it doesn't just ram them together. Instead it combines a copy of the second vector into each part of the first vector. In symbols:

$$(a,b)\otimes(x,y,z) \;=\; (a\cdot(x,y,z),\; b\cdot(x,y,z)) \;=\; (ax,ay,az,bx,by,bz).$$

The vectors we get have dimension 6 not 5. We will see that not every vector in the target space, here $\mathbb{R}^6$, arises as a tensor product. But if we close out $\mathbb{R}^2 \otimes \mathbb{R}^3$ under linear combinations, then we do get all of $\mathbb{R}^6$.

What does tensor product *do*? We feel again that good intuition comes by thinking about abstract attributes first, numbers later. Let us make the card suits into the abstract "attribute vector"

$$u \;=\; (\clubsuit, \diamondsuit, \heartsuit, \spadesuit).$$

And the ranks of cards becomes the attribute vector

$$v \;=\; (2,3,4,5,6,7,8,9,10,J,Q,K,A).$$

Then the tensor product---in the order $u \otimes v$ (it's not commutative)---is

$$w \;=\; (\clubsuit 2, \clubsuit 3, \;\ldots, \clubsuit K, \clubsuit A, \diamondsuit 2, \diamondsuit 3, \;\ldots, \diamondsuit A, \heartsuit 2, \;\ldots, \heartsuit A, \spadesuit 2, \;\ldots, \spadesuit A).$$

This sorts the deck by suits. If we tensored the other way around, we'd get

$$v \otimes u \;=\; (2\clubsuit, 2\diamondsuit, 2\heartsuit, 2\spadesuit, 3\clubsuit, 3\diamondsuit, 3\heartsuit, 3\spadesuit, 4\clubsuit, \;\ldots, 4\spadesuit, 5\clubsuit, \;\ldots, \;\ldots, A\clubsuit, A\diamondsuit, A\heartsuit, A\spadesuit),$$

which sorts the deck by ranks instead. Always the second vector gets "copied inner" while the first vector is "outer." The ordinary product would have just rammed v after u to give a vector of 17 items, four of type `Suit` and thirteen of type `Rank`. This is inhomogeneous mishmash---like "not playing with a full deck" as we say. Whereas, in either order, the tensor product creates a homogeneous length-52 vector of type "`Suit` and `Rank`." (Between `Suit` and `Rank`, the order might or might not matter.)

Now let's see how this works numerically. The vector $\mathbf{u} = [0,1,0,0]^T$ is the standard basis vector corresponding to "diamonds" in our scheme for suits. For ranks, the seven is indexed by

$$\mathbf{v} \;=\; [0,0,0,0,0,1,0,0,0,0,0,0,0].$$

Then

$$\begin{aligned}\mathbf{u}\otimes\mathbf{v} \;&=\; [0\cdot[0,0,0,0,0,1,0,0,0,0,0,0,0],\; 1\cdot[0,0,0,0,0,1,0,0,0,0,0,0,0],\; \ldots, 0]^T \\ &=\; [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,\; \ldots .0]^T.\end{aligned}$$

The single 1 corresponds to the position of the $\diamondsuit 7$ in the abstract indexing scheme. So we write:

$$\mathbf{u}\otimes\mathbf{v} = |\diamondsuit 7\rangle \;=\; |\diamondsuit\rangle|7\rangle.$$

Thus the tensor product of two standard basis vectors gives us a standard basis vector in the larger space. Indeed, we can get the entire standard basis of 52 vectors this way. We've also started writing the "invisible dot" product of kets---which are in quantum coordinates---to stand for the tensor product in the underlying coordinates.

Suppose we next take the tensor product of w with itself. Doing this with the attribute vectors, we get

$$w \otimes w \;=\; (\clubsuit 2 \clubsuit 2, \clubsuit 2 \clubsuit 3, \clubsuit 2 \clubsuit 4, \;\ldots, \clubsuit 2 \clubsuit A, \clubsuit 2 \diamondsuit 2, \;\ldots, \clubsuit 2 \spadesuit A, \;\clubsuit 3 \clubsuit 2, \clubsuit 3 \clubsuit 3, \;\ldots, \;\ldots, \spadesuit A \spadesuit A).$$

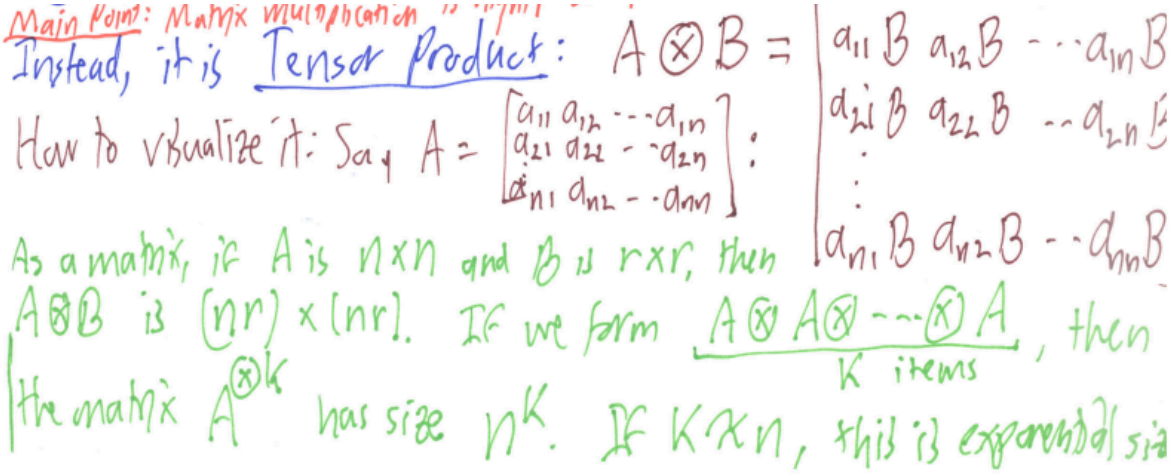
What does this represent? It conveys the idea of playing two cards in sequence. This allows them to be the same card---casinos usually play blackjack with eight decks shuffled together, for instance---but we will encounter algorithms whose point is either *amplifying* or eliminating such particular possibilities. The point is that tensor product is the underlying way of Nature simply doing a *sequence*, which means *concatenating* the symbolic representations.

Tensor Product = Simple Concatenation = Flow of Events.

This kind of representation is not just useful in quantum. It underlies the idea of the "TensorFlow" API and library in machine learning.

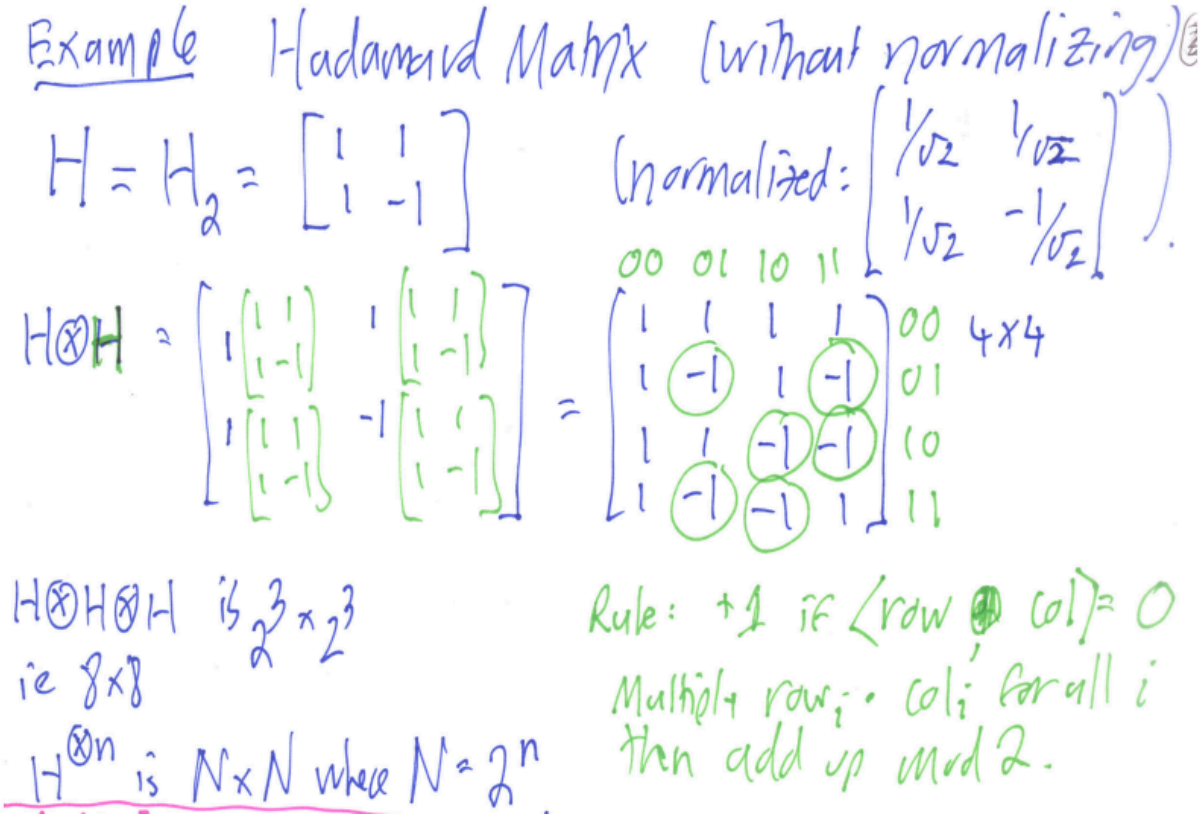
The common example that matters most is when both spaces have dimension 2. This case is innately confusing because $2 + 2 \;=\; 2 \cdot 2 \;=\; 2^2$. But hopefully the above will help us avoid confusion.

Tensor Products With Matrices



If A is $\ell \times m$ and B is $n \times r$ then $A \otimes B$ is $\ell n \times mr$, so the dimensions can be anything. In particular, A and B can both be column vectors with $m \;=\; r \;=\; 1$, whereupon $A \otimes B$ is a column vector of length ℓn.

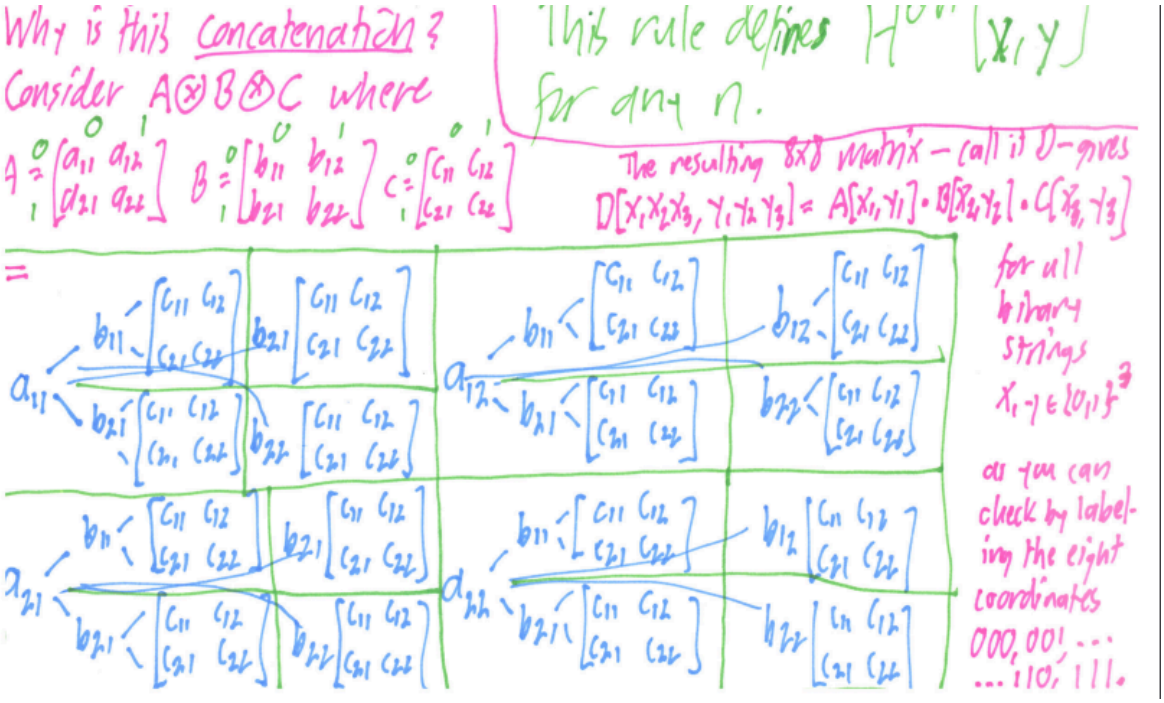
[Lecture first did an example of a 3×2 matrix A tensored with a 3×4 matrix B. It was tedious on-purpose. The resulting matrix C was 9×8. All entries of B were negative. One point of doing that was to distinguish the factors coming from each matrix---but even so, cases like $3 \cdot (-12)$ and $4 \cdot (-9)$ caused equal entries in C. This led to the question of whether tensor product of matrices is *lossy*---given C, can you get A and B back again, or not? Note that ordinary matrix multiplication $C = A \cdot B$ is mega-lossy (besides the fact that the dimensions must conform: $m = n$). A second point is that the negated entries could have been factored out as a -1 scalar outside the whole product, thus saving some scratchwork. This is true with any scalar factor---and this will soon be underscored by the definition that two quantum states (or two quantum operations) are equivalent if one is a multiple of the other by a scalar.]



The entries of $H^{\otimes n}$ are indexed by binary strings x, y of length n. Take the Boolean inner product mod 2 of x and y. If it is 0, then $H^{\otimes n}[x, y] = 1$, but if it is 1, then $H^{\otimes n}[x, y] = -1$.

E.g. $\langle 00, y \rangle = 0$ for any y, so the row for 00 is all 1s. But $\langle 01, 01 \rangle = 1$, so the entry $H^{\otimes 2}[01, 01] = -1$. And $\langle 11, 11 \rangle = 2 = 0 \ (mod\ 2)$, so $H^{\otimes 2}[11, 11] = +1$ back again.

This rule defines $H^{\otimes n}$ for any n as an $N \times N$ matrix. On paper that is exponential size, but in a **quantum circuit diagram** on n qubits, it is $O(n)$ gates. Is it linear effort for Nature to compute? Because the computation is unitary, hence **reversible** and ideally accompanied by zero entropy, it might be zero effort. Or, because it represents splitting beams of particles, possibly serially, it might be exponential effort after all.



Two Quantum Coordinates

In 4-space, the standard basis is given by the vectors:

$$e_0 = (1,0,0,0),\ e_1 = (0,1,0,0),\ e_2 = (0,0,1,0),\ e_3 = (0,0,0,1).$$

The indexing scheme for **quantum coordinates** changes the labels to come from $\{0,1\}^2$ instead of from $\{1,2,3,4\}$, using the canonical binary order $00, 01, 10, 11$. Then we have:

$$e_{00} = (1,0,0,0),\ e_{01} = (0,1,0,0),\ e_{10} = (0,0,1,0),\ e_{11} = (0,0,0,1).$$

The big advantage is that these basis elements are all separable and the labels respect the tensor products involved:

$$\begin{aligned} |00\rangle &= e_{00} = (1,0,0,0) = (1,0)\otimes(1,0) = e_0 \otimes e_0 = |0\rangle\otimes|0\rangle = |0\rangle|0\rangle \\ |01\rangle &= e_{01} = (0,1,0,0) = (1,0)\otimes(0,1) = e_0 \otimes e_1 = |0\rangle\otimes|1\rangle = |0\rangle|1\rangle \end{aligned}$$

$$|10\rangle = e_{10} = (0,0,1,0) = (0,1)\otimes(1,0) = e_1 \otimes e_0 = |1\rangle\otimes|0\rangle = |1\rangle|0\rangle$$
$$|11\rangle = e_{11} = (0,0,0,1) = (0,1)\otimes(0,1) = e_1 \otimes e_1 = |1\rangle\otimes|1\rangle = |1\rangle|1\rangle$$

It is OK to picture the tensoring with row vectors, but because humanity chose to write matrix-vector products as Mv rather than vM, they need to be treated as column vectors. This will lead to cognitive dissonance when we read quantum circuits left-to-right but have to compose matrices right-to-left.

We can also take tensor products of non-basis vectors of length 2. Let's us try

$$\mathbf{u} = \frac{e_0 + e_1}{\sqrt{2}} = \sqrt{0.5}\,[1,1]^T.$$

When we do $\mathbf{u}\otimes\mathbf{u}$, the first thing that happens is that the scalars in front multiply to get $\sqrt{0.5}\cdot\sqrt{0.5} = \frac{1}{2}$ as the multiplier on the whole thing. The vector bodies combine as

$$[1\cdot[1,1],\ 1\cdot[1,1]] = [1,1,1,1].$$

(Strictly speaking, we should do this as column vectors---maybe we'll show on the whiteboard---but it's always fine to do as row vectors and remember to transpose when needed at the end.) So

$$\mathbf{u}\otimes\mathbf{u} = \left[\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2}\right]^T.$$

This is a unit vector. We can do the same with

$$\mathbf{v} = \frac{e_0 - e_1}{\sqrt{2}} = \sqrt{0.5}[1,-1]^T$$

We get $\mathbf{v}\otimes\mathbf{v} = \frac{1}{2}$ times $[1,-1]\otimes[1,-1] = \frac{1}{2}[1,-1,-1,1]$. OK, $\frac{1}{2}[1,-1,-1,1]^T$ to be strict. We also get:

$$\mathbf{u}\otimes\mathbf{v} = \frac{1}{2}[1,-1,1,-1]^T.$$

$$\mathbf{v}\otimes\mathbf{u} = \frac{1}{2}[1,1,-1,-1]^T.$$

Suppose I instead said

$$\mathbf{u} \otimes \mathbf{v} \;=\; \frac{1}{2}[-1, 1, -1, 1]^T?$$

Would I be wrong? Numerically yes: $\mathbf{u} \otimes \mathbf{v}$ does not equal $-\mathbf{u} \otimes \mathbf{v}$. But Nature's "Quantum Rose" does not care. It *may*, however, care about the difference from $\mathbf{v} \otimes \mathbf{u}$.

Let's ignore the normalizing multipliers out front for the moment, since they do not matter to the ability to combine the vector bodies. How about the simpler vector

$$\mathbf{w} \;=\; [1, 0, 0, 1]?$$

This equals $e_{00} \;+\; e_{11}$, i.e., $|00\rangle + |11\rangle$, ignoring the normalizing constant $\sqrt{0.5}$. Can we get this as a tensor product of two vectors of length 2?

The answer is no. We can prove it by representing the general 2-by-2 tensor product as

$$[a, b] \otimes [c, d] \;=\; [ac, ad, bc, bd].$$

To get $[1, 0, 0, 1]$ as the result, we need to solve the equations

$$ac = 1,\; ad = 0,\; bc = 0, \text{ and } bd = 1.$$

But $ad = 0$ entails that either a is 0 or d is 0. If $a = 0$, then $ac = 1$ is impossible. But if $d = 0$, then $bd = 1$ is impossible. So there is no solution.

Definition. A vector is **separable** if it can be written as the tensor product of two smaller vectors. Otherwise---and especially when the vector represents a quantum state---we call it **entangled**.

To introduce some more quantum terminology, when a unit vector is not a basis vector, it is necessarily a linear combination of two or more basis vectors. Then it is a **superposition**. One of the amazing verities of physics is that we really can put particle-level sustems into superpositions and interact with them. The math of how those interactions behave involves the kind of vectors we are already seeing. The vectors $\mathbf{u}$ and $\mathbf{v}$ above are superpositions. When we re-interpret the attributes $|0\rangle$ and $|1\rangle$ as $|dead\rangle$ and $|alive\rangle$, then $\mathbf{u}$ becomes the superposition

$$\frac{|dead\rangle \;+\; |alive\rangle}{\sqrt{2}}$$

This is said to be the state of **Schrödinger's Cat**. The philosophical issue is whether a macro-level being, not a particle, can be put into superposition. (We will later argue the answer is "yes...but...") For now we prefer to take the simple realist view that a particle can have the state $\mathbf{u}$. It is not a cat, but it

has a pet-name, indeed a ket-name: $|+\rangle$. The vector $\mathbf{v}$, with its prominent minus sign, is called $|-\rangle$.

Tensoring the cat with another cat gave us $\frac{e_{00} + e_{01} + e_{10} + e_{11}}{2}$. Are the cats entangled? No---they are separable, because we got this as a tensor product to begin with.

Inner Products (begin Thu. 9/5)

The **dot-product** of two equal-length vectors $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots b_n)$ is always defined by

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + \cdots + a_n b_n.$$

(I actually want to write a bigger solid dot, and our textbook does so, but MathCha doesn't have it.) We have already seen the example of the dot product of binary strings treated as vectors with entries modulo 2, which is called the "Boolean inner product" on page 12 of Chapter 2.

When $\mathbf{a}$ and $\mathbf{b}$ are vectors of real-number entries, the **real inner product** is the same as the dot-product. Besides $\mathbf{a} \cdot \mathbf{b}$, the most common notation for it is $\langle \mathbf{a}, \mathbf{b} \rangle$.

When $\mathbf{a}$ and $\mathbf{b}$ are vectors over the *complex* numbers, however, the entries of the *first* vector are complex-conjugated. The common angle-bracket notation remains the same:

$$\langle \mathbf{a}, \mathbf{b} \rangle = \overline{a}_1 b_1 + \cdots + \overline{a}_n b_n.$$

This should not cause confusion with regard to the real-number case: if the entries a_i all happen to be real numbers, their conjugates don't change, so we get $a_1 b_1 + \cdots + a_n b_n$ anyway. But don't forget to conjugate when they really are complex numbers! The "twist" of conjugating the a_i entries is thus more fundamental. Any, this defines the "standard" inner product on a finite-dimensional real or complex vector space. (If you're curious about the general definition of when a vector space V becomes a **Hilbert Space**, it is when there is an abstract product $\cdot$ such that whenever you have a sequence $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots$ of vectors **in** the space and a vector $\mathbf{w}$ (that could be "floating outside") such that the scalar values $\mathbf{v}_1 \cdot \mathbf{w}, \mathbf{v}_2 \cdot \mathbf{w}, \mathbf{v}_3 \cdot \mathbf{w}, \ldots$ go to zero, the vector $\mathbf{w}$ in fact belongs to V.)

The conjugation twist may also seem a little arbitrary---why don't we conjugate the b_i entries instead? Well, this is where using a vertical bar rather than a comma leads to a notational inspiration by Paul Adrien Maurice Dirac in 1939.

$$\langle \mathbf{a} | \mathbf{b} \rangle = \overline{a}_1 b_1 + \cdots + \overline{a}_n b_n = \overline{\mathbf{a}} \cdot \mathbf{b}.$$

The first thing to note is that if we transpose $\overline{\mathbf{a}}$ to be a row-vector rather than a column vector, while

keeping $\mathbf{b}$ as a column vector (which you can call the default), then we can interpret the dot $\cdot$ as ordinary "matrix" multiplication. Rather than messily write the conjugate transpose as $\overline{\mathbf{a}}^T$, we write it as $\mathbf{a}^*$, where the superscript star can be read as "star", "adjoint", or "dual" depending on the context. (Word-to-the-wise, however: newer sources are using a superscript "dagger" instead: $\mathbf{a}^\dagger$. I avoid the dagger in handwriting because it can look either like a plus sign or like a superscript T for transpose without conjugating.) Now Dirac's inspiration was that we can break the angled "bracket" into two pieces at the vertical bar:

$$\langle \mathbf{a}|\mathbf{b}\rangle \;=\; \langle \mathbf{a}| \cdot |\mathbf{b}\rangle \;=\; \mathbf{a}^* \cdot \mathbf{b}$$

where the dot is the same as our dot product and ordinary "matrix" multiplication. He called the two parts the "bra" and the "ket". Thus the bra $\langle \mathbf{a}|$ is another way of writing $\mathbf{a}^*$. (And when a is a scalar, a^* is another way of writing $\overline{a}$ since transpose is immaterial.) It comes IMHO with a different philosophy: instead of the "pristine" vector $\mathbf{a}$ being changed by conjugation, it is "moved into dual position." It is considered OK to use "$\mathbf{a}$" and "$|\mathbf{a}\rangle$" as synonymous notations, although we've already used the ket notation as wrapper around an *attribute*, to signify the basis state indexing that attribute. Well, keeping things simple is why Part I of the text avoids the Dirac notation, but I think conveying Dirac's insight will help tie our various "product" ideas together: We can write bras and kets together in products any which way. If we write

$$|\mathbf{a}\rangle \cdot |\mathbf{b}\rangle$$

then this is the tensor product of the two vectors. We've already been doing this with our standard basis states of binary-string attributes under big-endian notation, e.g. (making the dot $\cdot$ invisible):

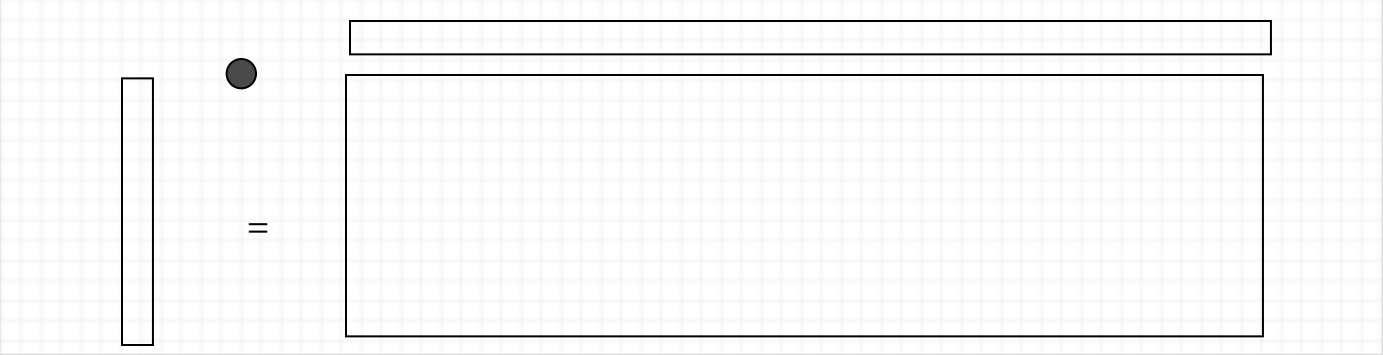
$$|0\rangle|1\rangle \;=\; |0\rangle \otimes |1\rangle \;=\; e_0 \otimes e_1 \;=\; e_{01} \;=\; |01\rangle$$
$$|10\rangle|11\rangle \;=\; |10\rangle \otimes |11\rangle \;=\; e_{10} \otimes e_{11} \;=\; e_{1011} \;=\; |1011\rangle$$

..

And then $\langle \mathbf{a}| \cdot \langle \mathbf{b}|$ is the tensor product of the corresponding row vectors and works out the same as $\big(|\mathbf{a}\rangle \cdot |\mathbf{b}\rangle\big)^*$ owing to the scalar conjugation rule $\overline{a} \cdot \overline{b} \;=\; \overline{ab}$ applied to each entry. Finally,

$$|\mathbf{a}\rangle \cdot \langle \mathbf{b}|$$

can be figured out as multiplying an $\ell \times 1$ column vector by a (conjugated) $1 \times n$ row vector. You get an $\ell \times n$ matrix. Here is a sketch picture:



It actually has the same entries $a_i b_j^*$ as $\mathbf{a} \otimes \mathbf{b}^*$, but rolled into a matrix rather than left as a longer vector. Especially when $\ell = n$, this is called the **outer-product** matrix. And the final secret is that when $\mathbf{b} = \mathbf{a}$, the outer-product matrix

$$\mathbf{A} = |\mathbf{a}\rangle \cdot \langle \mathbf{a}|$$

captures all the ways that the quantum state $|\mathbf{a}\rangle$ can interact with the outside world. Most in particular, when we take the ordinary matrix product of $\mathbf{A}$ with another quantum state $|\mathbf{c}\rangle$ of compatible size, the associativity of basic dot multiplication gives us:

$$\mathbf{A} \cdot \mathbf{c} = \big(|\mathbf{a}\rangle \cdot \langle \mathbf{a}|\big) \cdot |\mathbf{c}\rangle = |\mathbf{a}\rangle \cdot \big(\langle \mathbf{a}|\mathbf{c}\rangle\big),$$

which is just the original vector $\mathbf{a}$ multiplied by the scalar value $\langle \mathbf{a}|\mathbf{c}\rangle$ from the inner product. The essence $|\mathbf{a}\rangle \cdot \langle \mathbf{a}|$ has a neat symmetry that maybe requites the concern about making a left-handed versus right-handed choice with conjugation. Maybe Nature is "evenhanded" after all. Moreover, this shows that all of our products:

- "ordinary" matrix product
- inner product
- outer product, and
- tensor product

are really ***fungible***---and belong to a reality where we can climb up a "stairway to heaven" of higher dimensions and back down again, so that maybe dimensionality is not fundamental. But the high-volume ThePhysicsMemes Twitter site recently had the last word.

Unit Vectors and Unitary Operations

Back on ground level, note that for a complex scalar c, c^*c and cc^* both equal its squared magnitude $|c|^2$. To wit, if we write $c = a + bi$ with a and b real, then $c^* = a - bi$, and we get

$$cc^* \;=\; (a+bi)(a-bi) \;=\; a^2 \;-abi \;+\; abi \;+\; b^2 \;=\; a^2+b^2 \;=\; |c|^2.$$

And for a vector **a**,

$$\langle \mathbf{a}|\mathbf{a}\rangle \;=\; \sum_{i=1}^{n} a_i^* a_i \;=\; \sum_{i=1}^{n} |a_i|^2$$

is its **squared magnitude**. The **norm** of the vector is the positive square root of this:

$$|\mathbf{a}| \;=\; \sqrt{\langle \mathbf{a}|\mathbf{a}\rangle}.$$

Definition: A **unit vector** has norm 1, equivalently, has squared magnitude 1. Any unit vector $\mathbf{v}$ in a Hilbert space gives rise to a "**legal quantum state**", which we can write as $|\mathbf{v}\rangle$ for pretentious emphasis (or decide not to do so).

In the two-dimensional real space $\mathbb{R}^2$, the unit vectors (x, y) correspond to the points on the unit circle in the plane, as defined by $x^2 + y^2 = 1$. The simplest quantum states are unit vectors of two *complex* numbers, i.e., members of $\mathbb{C}^2$. But in many cases we can ignore the distinction between complex and real entries and pretend-visualize them as points on the unit circle in the plane anyway. (There is a non-fudged, non-lossy visualization in 3D called the **Bloch Sphere**---we will come to it later.)

Now for operations, we begin by extending some notation we just used for vectors:

Definition. The **conjugate transpose** of an $m \times n$ matrix A is obtained by first transposing A to make the $n \times m$ matrix A^T, and then taking the complex conjugate of every element. We write A^* for the resulting $n \times m$ matrix. (Many other sources write $A^\dagger$ instead.)

Examples:

- If $A = \begin{bmatrix} 1+i & 1-i \\ 2 & 0 \end{bmatrix}$, then $A^T = \begin{bmatrix} 1+i & 2 \\ 1-i & 0 \end{bmatrix}$ and $A^* = \begin{bmatrix} 1-i & 2 \\ 1+i & 0 \end{bmatrix}$.
- If $\mathbf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$, then $\mathbf{Y}^T = \begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix}$, and weirdly, $\mathbf{Y}^* = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \mathbf{Y}$ back again.
- For our old friend the Hadamard matrix $\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $\mathbf{H}^T = \mathbf{H}^* = \mathbf{H}$ because $\mathbf{H}$ is symmetric and has all-real entries.

Note that

$$\mathbf{H}e_0 \;=\; \mathbf{H}|0\rangle \;=\; \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \;=\; \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \;=\; \frac{e_0 + e_1}{\sqrt{2}},$$

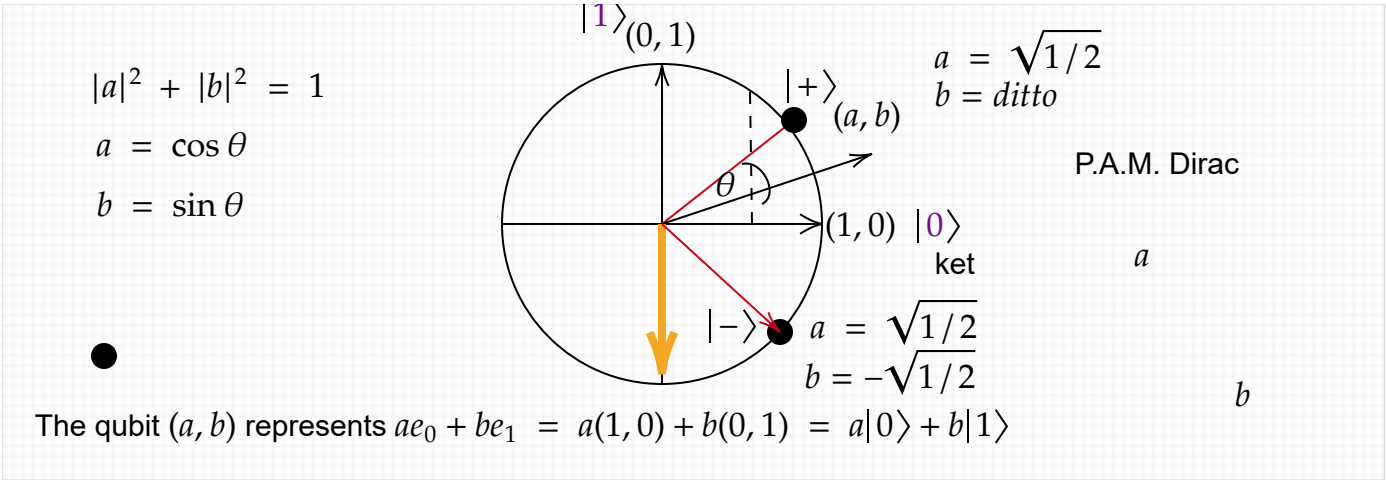
a vector we saw in the previous lecture, and ditto for

$$\mathbf{H}e_1 \;=\; \mathbf{H}|1\rangle \;=\; \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \;=\; \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} \;=\; \frac{e_0 - e_1}{\sqrt{2}}\,.$$

The two resulting vectors are also unit vectors, and they are linearly independent. Indeed, they are **orthogonal**, because

$$\langle [1,1],\ [1,-1] \rangle \;=\; 1\cdot 1 \;+\; 1\cdot(-1) \;=\; 0.$$

Thus, like e_0 and e_1, they form an **orthonormal basis** for $\mathbb{C}^2$ (not to mention also for $\mathbb{R}^2$). Focusing on the plus and minus sign between e_0 and e_1, the "Dirac names" for these states are $|+\rangle$ and $|-\rangle$, respectively. Getting just a little bit ahead for visualizing qubits, here is a diagram:



Well, $|+\rangle$ was our "Schrödinger's Cat" state where we spoke of **superposition**. Maybe that seemed mysterious. Now $\mathbf{H}$ carries the standard basis onto the $|+\rangle, |-\rangle$ basis. It also maps that basis back to the standard one, because $\mathbf{H}^2 \;=\; \mathbf{I}$, the 2×2 identity matrix. Thus, a vector that looks superposed in the standard basis can be simple when viewed in the changed basis. Thus superposition is relative---"in the eye of the beholder" one might say---but in many concrete cases the observer is Nature.

The matrix $\mathbf{Y}$ is one of four named after the quantum phsyicist Wolfgang Pauli. The others are

$$\mathbf{X} \;=\; \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{Z} \;=\; \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

and the identity $\mathbf{I}$. Note that $\mathbf{X}|0\rangle \;=\; \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \;=\; \begin{bmatrix} 0 \\ 1 \end{bmatrix} \;=\; |1\rangle$ and similarly, $\mathbf{X}|1\rangle = |0\rangle$. Thus applying $\mathbf{X}$ negates the bit label of a standard basis state, and this functions just like the Boolean NOT operation. Moreover, $\mathbf{X}$ is a **permutation matrix**. In upcoming lectures we will show how permutation matrices used in quantum circuits confer exactly the power of classical Boolean circuit gates. The extra

quantum power starts coming in with the Hadamard gate.

Note: $\mathbf{HZH}^{-1} = \mathbf{HZH} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} = \mathbf{X}$

Now for two key definitions (which apply to any size matrices, not just 2×2):

Definition: A matrix A is **unitary** if $A^*A = \mathbf{I}$.

Note, incidentally, that A must be invertible, and furthermore

$$AA^* = AA^*\left(AA^{-1}\right) = A\left(A^*A\right)A^{-1} = A\mathbf{I}A^{-1} = AA^{-1} = \mathbf{I}.$$

This also works vice-versa: if $AA^* = \mathbf{I}$, then $A^*A = \mathbf{I}$. So an equivalent definition of unitary is that $AA^* = \mathbf{I}$.

Definition: A matrix A is **Hermitian** if $A^* = A$.

The Pauli matrices are all both Hermitian and unitary. So is the Hadamard matrix.

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \mathbf{I}.$$

If we took away the factor $\frac{1}{\sqrt{2}}$, the resulting matrix $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ is Hermitian but not unitary. The matrix $\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ is unitary but not Hermitian.

In Part I of the text we toe the line of identifying unitary matrices with "legal quantum operations." When we dabble into Chapter 14, we will encounter the view that Hermitian operators are the "physically actual" ones. Most in particular:

Proposition: For any unit vector $\mathbf{c}$, the outer product $C = |\mathbf{c}\rangle\langle\mathbf{c}|$ is a Hermitian matrix.

Proof: It is a general fact that if $C = AB$, then $C^* = (AB)^* = B^*A^*$. So

$$C^* = \left(|\mathbf{c}\rangle\langle\mathbf{c}|\right)^* = \left(\langle\mathbf{c}|\right)^* \cdot \left(|\mathbf{c}\rangle\right)^* = |\mathbf{c}\rangle \cdot \langle\mathbf{c}| = C$$

back again. ☒

Now we can use the reversal rule for adjoints to give a shorter and snappier proof of Lemma 3.1 than what the text gives:

Lemma 3.1: If $\mathbf{U}$ is a unitary matrix and $\mathbf{a}$ is a vector then $||\mathbf{Ua}|| = ||\mathbf{a}||$.

Proof: $||\mathbf{Ua}|| = \sqrt{||\mathbf{Ua}||^2} = \sqrt{(\mathbf{Ua})^*(\mathbf{Ua})} = \sqrt{(\mathbf{a^*U^*})(\mathbf{Ua})} = \sqrt{\mathbf{a}^*(\mathbf{U^*U})\mathbf{a}} = \sqrt{\mathbf{a}^*\mathbf{a}} = ||\mathbf{a}||.\boxtimes$

The proof became a one-liner. Thus a unitary matrix always preseves the lengths of vectors, and in particular, it always maps a unit vector to a unit vector. This is what makes it "legal" from the quantum probability point of view.

The fact works the other way: if a matrix U always preserves the lengths of vectors, then it must be unitary.

Reversal, Adjoint, and Duality.

The reversal x^R of a string x just means writing it "backwards": $01001^R = 10010$, FACED^R = DECAF, and so on. A string x is a palindrome if $x^R = x$, for instance 1001. The empty string ϵ counts as a palindrome since $\epsilon^R = \epsilon$. The rule for reversal and concatenation is that for any strings x and y,

$$(xy)^R = y^R x^R.$$

For example,

$$\text{(PUCK-FACED)}^R = (\text{FACED})^R(\text{PUCK-})^R = \text{DECAF-KCUP}.$$

Actually, if the minus sign is a -1 factor which could go anywhere, this would be equivalent to say "DECAF K-CUP" meaning a certain pod for a Keurig coffee-maker.

This gives intuition for how matrix transpose, matrix adjoint, and matrix inverse all work like reversal with regard to matrix product. The rules for any (invertible) matrices A and B are:

1. $(AB)^T = B^T A^T$
2. $(AB)^* = B^* A^*$
3. $(AB)^{-1} = B^{-1} A^{-1}$.

Rule 2 follows from rule 1 because the only difference with * is doing complex conjugates of individual entries. Rule 3 follows since $(AB)(B^{-1}A^{-1}) = ABB^{-1}A^{-1} = AA^{-1} = \mathbf{I}$. So why does rule 1

hold? Here our functional view might help: The transpose A^T is the function with the two index arguments reversed: $A^T(j,i) = A(i,j)$. So:

$$(AB)^T(i,j) = (AB)(j,i) = \sum_k A(j,k)B(k,i) = \sum_k B(k,i)A(j,k) = \sum_k B^T(i,k)A^T(k,j) = B^TA^T(i,j)$$

for all arguments (i.e., indices) i and j, so $(AB)^T = B^TA^T$. (Note that the switch $A(j,k)B(k,i) = B(k,i)A(j,k)$ in the middle step was just ordinary multiplication of numbers.)

The adjoint $\mathbf{x}^*$ of a vector $\mathbf{x}$ has another interpretation. It stands ready to pounce on any column vector $\mathbf{y}$ of the same length as $\mathbf{x}$ and wrangle it down to the scalar

$$\mathbf{x}^*\mathbf{y} = \sum_i \overline{\mathbf{x}[i]}\mathbf{y}[i] = \langle \mathbf{x}, \mathbf{y} \rangle,$$

which is the inner product of $\mathbf{x}$ and $\mathbf{y}$. As such, $\mathbf{x}^*$ defines the **linear functional** $f_{\mathbf{x}} : \mathbb{H}^n \to \mathbb{H}$ by

$$f_{\mathbf{x}}(\mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle.$$

Whereas a column vector is to be interpreted as "data", the row-vector form is "code".

Now we can review and extend some of the things we said about Dirac notation and the relationships between various kinds of products:

1. If $\mathbf{z} = a\mathbf{x}$ where $\mathbf{x}$ and $\mathbf{z}$ are numeric vectors and a is a (possibly complex) scalar, then we have the rule $\mathbf{z}^* = \overline{a}\mathbf{x}^*$. We have to remember to conjugate any factor we pull out of the adjoint. A Khan Faculty video writes the rules $|a\psi\rangle = a|\psi\rangle$ and $\langle a\psi| = a^*\langle\psi|$, but you have to be careful that ψ stands for a numeric vector here. It makes no sense to say e.g. that $3|1\rangle = |3\rangle$ when the $|1\rangle$ is the binary-bit attribute, nor that $3|7\rangle = |21\rangle$ if the "7" is the rank of a playing card. (Note that it is more convenient to write a^* rather than $\overline{a}$ for the complex conjugate of a scalar, as if it were a "1×1" dimensioned entity. And scalar multiplication commputes, so writing $(ab)^* = b^*a^*$ equates to a^*b^*, in accordance with the rule $\overline{ab} = \overline{a} \cdot \overline{b}$.)
2. $\langle x| \cdot |y\rangle = \langle x|y\rangle$. The product dot first goes invisible, then the two vertical bars combine to be one.
3. $\langle y|x\rangle = \langle y| \cdot |x\rangle = |y\rangle^* \cdot \langle x|^* = \big(\langle x| \cdot |y\rangle\big)^* = \langle x|y\rangle^*$ by the reversal rule. So the flipped-around inner product $\langle y|x\rangle$ is just the complex conjugate of the scalar $\langle x|y\rangle$.
4. Two consecutive kets as in $|x\rangle|y\rangle$ is a gray area. Equating it to $|x\rangle \otimes |y\rangle$ is AOK when manipulating standard basis vectors, e.g. $|1\rangle|0\rangle|0\rangle|1\rangle|0\rangle = |10010\rangle$. Likewise,

$$|+\rangle|+\rangle = \frac{1}{\sqrt{2}}[1,1] \otimes \frac{1}{\sqrt{2}}[1,1] = \frac{1}{2}[1,1,1,1]^T = |{+}{+}\rangle$$

is kosher as nomenclature, likewise writing $|+\rangle|-\rangle$ as $|+-\rangle$ and so on. But the product of two column vectors is not really defined, and in general cases of $|x\rangle|y\rangle$ where "x" and "y" are *not* what I have been calling "attributes", combo-ing it as "$|xy\rangle$" may not make sense. What might bail you out of doubt is if you have a bra $\langle w|$ before $|x\rangle|y\rangle$. Then it becomes $\langle w|x\rangle \cdot |y\rangle$, where the $\cdot$ is now the same as ordinary multiplication. But $\langle w| \cdot |xy\rangle$ may not make sense off the top---because inner product wants the dimensions to agree. (??)

5. Two consecutive bras like $\langle x|\langle y|$ are even grayer. Would they be the adjoint of $|y\rangle|x\rangle$ or of $|x\rangle|y\rangle$? Note what happens for tensor products of matrices: For all indices u, v, w, t,

$$(A \otimes B)^*(uv, wt) \;=\; \overline{(A \otimes B)(wt, uv)} \;=\; \overline{A(w,u)B(t,v)} \;=\; \overline{A(w,u)} \cdot \overline{B(t,v)}$$
$$=\; A^*(u,w)B^*(v,t) \;=\; \big(A^* \otimes B^*\big)(uv, wt).$$

So $(A \otimes B)^* \;=\; A^* \otimes B^*$. Did you expect the A and B to reverse? Maybe not if you realize that they operate in independent systems.

[This last point 5 was actually meant to come after the "Operations: Joint and Entangled" subsection below. So I will revisit it early in the next lecture.]