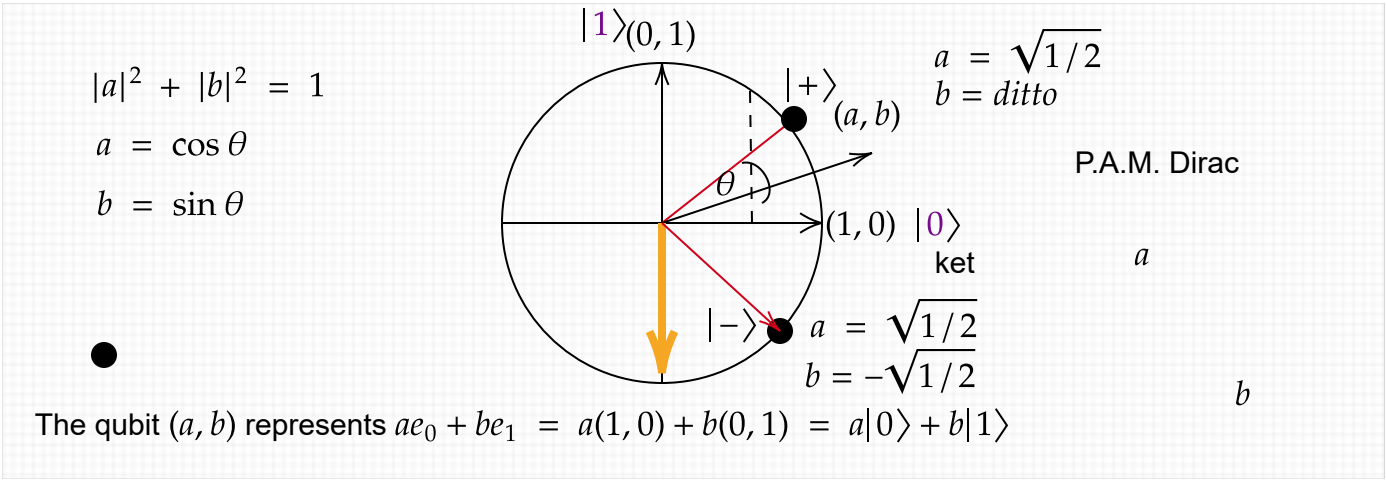
CSE439/510 Week 3: Operations on Quantum States and Computations (chs. 3 into 4)

The first part has been seen before. In Fall 2025 I zipped over it but reminded how we showed $\mathbf{H}e_0 = \frac{1}{\sqrt{2}}[1,1]^T$, which we called $|+\rangle$, and $\mathbf{H}e_1 = \frac{1}{\sqrt{2}}[1,-1]^T$, which we called $|-\rangle$ because of the $-$ sign. The Hadamard matrix thus does a change of basis operation---from the standard basis to the $|+\rangle, |-\rangle$ basis, which is also an **orthonormal** basis. Also, I did parts on the blackboard that are reviewed in the section of "reversals and..." below.

Unitary Operations

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 $\mathbf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ 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 $AA^* = \mathbf{I}$.

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

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.

[In Fall 2025, I covered the meat of this a different way on the blackboard, so will skim here.]

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,

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

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 $(AB)\big(B^{-1}A^{-1}\big) = ABB^{-1}A^{-1} = AA^{-1} = \mathbf{I}$. 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:

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

for all arguments (i.e., indices) i and j, so $(AB)^T = B^T A^T$. (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.)

Also note that transpose, reversal, inverse, and adjoint all have the "self-dual" property that applying them twice gives the original back again: $\left(x^R\right)^R = x$, $\left(A^T\right)^T = A$, $\left(A^{-1}\right)^{-1} = A$, and $\left(A^*\right)^* = A$.

Now we can review some other basic useful rules, with-and-without Dirac notation:

1. Dot product **is** commutative: $x \cdot y = y \cdot x$. This is implicit in the step in red above.
2. Complex inner product is "not quite" commutative:
   $\langle y|x\rangle = \langle y| \cdot |x\rangle = |y\rangle^* \cdot \langle x|^* = \left(\langle x| \cdot |y\rangle\right)^* = \langle x|y\rangle^*$ by the reversal rule. So the flipped-around inner product $\langle y|x\rangle$ is just the complex conjugate of the scalar $\langle x|y\rangle$. But often the difference caused by the conjugate is minimal.
3. Outer product is not commutative. But: $\left(|y\rangle\langle x|\right)^* = \langle x|^*|y\rangle^*$ by the reversal rule, and by the definitions of "ket" and "bra" that becomes $|x\rangle\langle y|$. So flipping the outer product around creates the adjoint of the matrix. This is another reason why taking the outer product of a vector with itself always creates a self-adjoint (i.e., **Hermitian**) matrix.
4. Tensor product is not commutative: generally $A \otimes B \neq B \otimes A$. But here's a real curveball: $(A \otimes B)^*$ is **NOT** the same as $\left(B^* \otimes A^*\right)$. Instead, it equals $\left(A^* \otimes B^*\right)$. The intuitive reason, which we'll picture soon when we visualize quantum circuits, is that elements of tensor products "stay in their lane" when it comes to the indexing scheme.

All of these products are, however, *linear on both sides*. That is, they obey the distributive law for addition on either side, and (hence) constant multiples also can "come out" or "go inside". Here are the cases for linearity on the right-hand side, assuming the vectors and matrices being added have compatible dimensions and a is a (possibly complex) scalar:

- $u \cdot (av + w) = a(u \cdot v) + u \cdot w$.
- $\langle u|av + w\rangle = a\langle u|v\rangle + \langle u|w\rangle$.
- But note on the left: $\langle au + v|w\rangle = a^*\langle u|w\rangle + \langle v|w\rangle$, because a is implicitly conjugated "inside the bra".
- And furthermore, this time on the right: $|u\rangle\langle av + w| = a^*|u\rangle\langle v| + |u\rangle\langle w|$. Same reason coming from inside the "bra". This is a common cause of typos.
- $A \cdot (aB + C) = aA \cdot B + A \cdot C$.
- $A \otimes (aB + C) = a(A \otimes B) + (A \otimes C)$.

- $(aC)^* = C^*a^* = a^*C^*$ because scalars commute with matrices. Ditto with vectors. Using bar instead of star: 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.

Super-weeny point: 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. This is one reason I used card-suit symbols and "honor/spot-card", to get away from the temptation of treating the insides of $|0\rangle$ and $|1\rangle$ as if they were numbers rather than attributes.

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. (??)

Unitary Versus Stochastic (section 3.6)

A (doubly) **stochastic** matrix has the property that its rows (and columns) are nonnegative real numbers that sum to 1. A simple example is

$$J = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

However, while J is Hermitian (like any symmetric real matrix), it is not unitary: $JJ^* = J^2 = J$, not the identity. There are doubly stochastic matrices that are not Hermitian either when we go up to 3×3, e.g.:

$$\begin{bmatrix} 1/2 & 1/3 & 1/6 \\ 1/2 & 1/6 & 1/3 \\ 0 & 1/2 & 1/2 \end{bmatrix}$$

However, every **permutation matrix** is both doubly stochastic (in the trivial manner of having a single 1 in each row and column) and unitary. A less trivial example of symmetric (Hermitian) doubly stochastic

matrices arise from **undirected graphs** G that are **regular**---meaning every vertex in G has the same **degree** (meaning: number of edges connecting to it). The text in section 3.6 gives an example where negating some of the entries does create a unitary matrix. However, this is not a regular phenomenon as far as I know.

The upshot of all this is:

- Legal quantum states are identified with *unit vectors*.
- Legal quantum operations are identified with *unitary matrices*.

Now we will define computations using these objects.

Quantum Computations

[The flow of Chapter 4 as written is to take the classical notion of computations by machines as given. When CSE396 was a required course at UB, everyone saw *Turing machines* (TMs); those may have been talked about briefly in CSE331, but otherwise the "random-access machine" concept of executing algorithms from that course is fine. (The one advantage of TMs is that you can say that their tape cells numbered 1,2,3,... represent "classical bits" that evolve over time, in analogy to the way we will speak of *qubits* evolving over time.) Now, however, we will take the *classical Boolean circuit* model as fundamental while contrasting it directly to quantum circuits. The strongest linkage is that the quantum **Toffoli gate** can simulate NAND and hence do all classical Boolean operations by itself. This is shown in section 5.3, though. Section 5.1 has the n-fold tensor product $\mathbf{H}^{\otimes n}$ of the basic 2×2 Hadamard matrix $\mathbf{H}$, which we have already seen, anyway. So please read all the above as one block.]

Operations: Joint and Entangled

Here is a statement that uses a lot of notational fuss to express the simplest of ideas:

Proposition: For any $m \times n$ matrix A, $p \times q$ matrix B, n-vector $\mathbf{x}$ and q-vector $\mathbf{y}$,

$$(A\mathbf{x}) \otimes (B\mathbf{y}) \;=\; (A \otimes B) \cdot (\mathbf{x} \otimes \mathbf{y}).$$

Proof. The dimensions are consistent: both sides give a column vector of mp entries. Showing equality is where our effort to interpret vectors $\mathbf{x}$ as functions $\mathbf{x}(u)$ of their indices in binary notation may help. Under this view, $\mathbf{z} \;=\; \mathbf{x} \otimes \mathbf{y}$ gives the function $\mathbf{z}(uv) \;=\; \mathbf{x}(u)\mathbf{y}(v)$, where uv means concatenation of binary strings, while the right-hand side is an ordinary numeric product. And a matrix A gives the two-argument function $A(u, w) \;=\; a_{u,w}$.

$[0.5, 0.5, -0.5, 0.5] \otimes [0.6, 0.8] \;=\; [0.3, 0.4, 0.3, 0.4, -0.3, -0.4, 0.3, 0.4]$
Indices: $[000, 001, 010, 011, 100, 101, 110, 111]$ $100 = 10 \cdot 0$: $-0.3 \;=\; (-0.5)(0.6)$.

Silly? style note: When we think of vector and matrix entries the way we usually do, we will use square brackets like in the text, e.g.: $\mathbf{x}[i]$, $\mathbf{A}[i, j]$. When the indices are regarded as binary strings rather than numbers, we will write things like $\mathbf{A}[u, w]$ and $\mathbf{C}[uv, wt]$ below, where $\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$.

The vector $\mathbf{x}' = A\mathbf{x}$ becomes the function mapping a row-index u to $\mathbf{x}'(u) = \sum_w A(u, w)\mathbf{x}(w)$. Thus, putting $\mathbf{z}' = (A\mathbf{x}) \otimes (B\mathbf{y})$, the right-hand side is the function

$$\mathbf{z}'(uv) = \mathbf{x}'(u)\mathbf{y}'(v) = \left(\sum_w A(u, w)\mathbf{x}(w)\right) \cdot \left(\sum_t B(v, t)\mathbf{y}(t)\right)$$

Now by usual rules of re-ordering summations, the right-hand side of this can be rearranged as

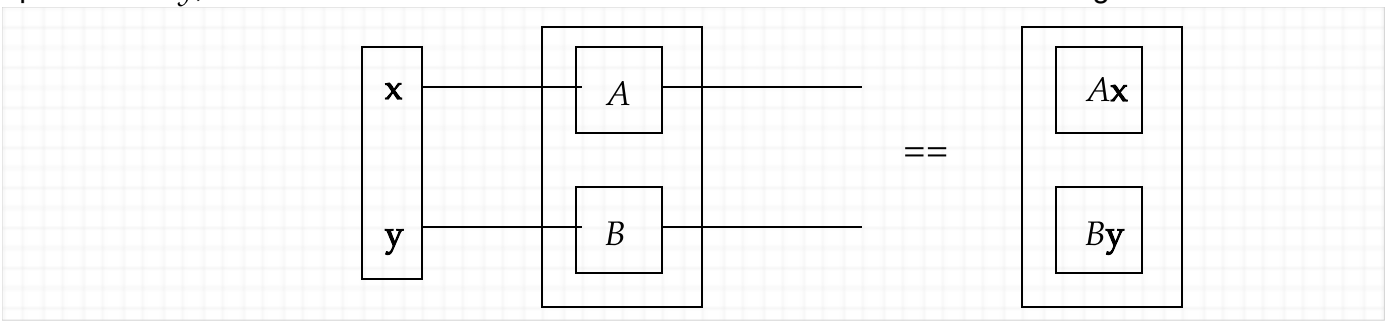
$$\sum_w \sum_t A(u, w)B(v, t)\mathbf{x}(w)\mathbf{y}(t)$$

With $\mathbf{z} = \mathbf{x} \otimes \mathbf{y}$, we can already recognize that the $\mathbf{x}(w)\mathbf{y}(t)$ part is the same as $\mathbf{z}(wt)$. And $A(u, w)B(v, t)$ is the same as $(A \otimes B)(uv, wt)$. So the whole thing becomes

$$\sum_{w,t} (A \otimes B)(uv, wt) \cdot (\mathbf{x} \otimes \mathbf{y})(wt),$$

which is exactly the meaning of $(A \otimes B) \cdot (\mathbf{x} \otimes \mathbf{y})$. So the two sides are equal. ☒

The simple idea is that $(A \otimes B) \cdot (\mathbf{x} \otimes \mathbf{y})$ does the A operation on x side-by-side with B doing its operation on y, but with no connection at all between them. We will soon have diagrams like this---



---note that we picture the inputs coming in from the left but when writing them as matrix arguments they will swing around to the right. As a tandem, this is formally the tensor product $\mathbf{x} \otimes \mathbf{y}$ coming in to $(A \otimes B)$. But really---and **locally**---it is just $A\mathbf{x}$ happening in one place and $B\mathbf{y}$ happening independently in another place. The upshot is this:

> **When we have entanglement, not independence, between the $\mathbf{x}$ part and the $\mathbf{y}$ part, then the notation will stay the same but the interpretation will change a whole lot.**

Multi-Qubit Matrices and Gates

The tensor product of two basic Hadamard gates is

$$\mathbf{H}^{\otimes 2} = \mathbf{H} \otimes \mathbf{H} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right].$$

This matrix carries the orthonormal two-qubit standard basis $e_{00}, e_{01}, e_{10}, e_{11}$ onto the four combinations of tensoring the $|+\rangle$ and $|-\rangle$ states, namely (transpose $\{\}^T$ omitted):

$$\begin{aligned}
|{+}{+}\rangle &= |+\rangle \otimes |+\rangle = \tfrac{1}{2}(1,\ 1) \otimes (1,\ 1) = \tfrac{1}{2}(1,\ 1,\ 1,\ 1) = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} \\
|{+}{-}\rangle &= |+\rangle \otimes |-\rangle = \tfrac{1}{2}(1,\ 1) \otimes (1,-1) = \tfrac{1}{2}(1,-1,\ 1,-1) = \frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2} \\
|{-}{+}\rangle &= |-\rangle \otimes |+\rangle = \tfrac{1}{2}(1,-1) \otimes (1,\ 1) = \tfrac{1}{2}(1,\ 1,-1,-1) = \frac{|00\rangle + |01\rangle - |10\rangle - |11\rangle}{2} \\
|{-}{-}\rangle &= |-\rangle \otimes |-\rangle = \tfrac{1}{2}(1,-1) \otimes (1,-1) = \tfrac{1}{2}(1,-1,-1,\ 1) = \frac{|00\rangle - |01\rangle - |10\rangle + |11\rangle}{2}
\end{aligned}$$

These four vectors are linearly independent and mutually orthogonal, so they form an orthonormal basis. We can see the mapping because forming the target vectors into a matrix (as column vectors) gives us exactly $\mathbf{H}^{\otimes 2}$.

Well, this is the case $m = 2$ of the Hadamard transform $\mathbf{H}^{\otimes m}$. Also note the following tensor products of 2×2 matrices:

$$\mathbf{H} \otimes \mathbf{I} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{array}\right],$$

$$\mathbf{I} \otimes \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}}\left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right].$$

Some examples of states you can produce with these matrices are:

$$\begin{aligned}
|{+}0\rangle &= |+\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}}(1,1) \otimes (1,0) = \frac{1}{\sqrt{2}}(1,0,1,0) = \frac{|00\rangle + |10\rangle}{\sqrt{2}} \\
|0{+}\rangle &= |0\rangle \otimes |+\rangle = \frac{1}{\sqrt{2}}(1,0) \otimes (1,1) = \frac{1}{\sqrt{2}}(1,1,0,0) = \frac{|00\rangle + |01\rangle}{\sqrt{2}}
\end{aligned}$$

Meanwhile,

$$|{+}1\rangle \;=\; |{+}\rangle \otimes |1\rangle \;=\; \frac{1}{\sqrt{2}}(1,1)\otimes(0,1) \;=\; \frac{1}{\sqrt{2}}(0,1,0,1) = \frac{|01\rangle + |11\rangle}{\sqrt{2}}$$

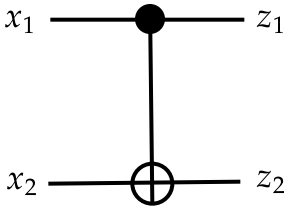
can be gotten as $\mathbf{H} \otimes \mathbf{I}$ applied to the column vector $(0,1,0,0)^T \;=\; |01\rangle$. However, the state $\frac{1}{\sqrt{2}}(1,0,0,1) \;=\; \frac{|00\rangle + |11\rangle}{\sqrt{2}}$, which we saw in the last lecture is entangled, cannot be gotten this way. Instead, it needs the help of a 4×4 unitary matrix that is not a tensor product of two smaller matrices. The most omnipresent one of these is:

$$\mathbf{CNOT} \;=\; \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Any linear operator is uniquely defined by its values on a particular basis, and on the standard basis, the values are: $\mathbf{CNOT}e_{00} \;=\; \mathbf{CNOT}|00\rangle \;=\; |00\rangle$, $\mathbf{CNOT}e_{01} \;=\; \mathbf{CNOT}|01\rangle \;=\; |01\rangle$, $\mathbf{CNOT}e_{10} \;=\; \mathbf{CNOT}|10\rangle \;=\; |11\rangle$, and $\mathbf{CNOT}e_{11} \;=\; \mathbf{CNOT}|11\rangle \;=\; |10\rangle$. We can get these from the respective columns of the $\mathbf{CNOT}$ matrix, and we can label the quantum coordinates right on it:

$$\mathbf{CNOT} \;=\; \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

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



	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	0	1
11	0	0	1	0

Because we multiply column vectors, the co-ordinates of the argument vector come in the top and go out to the left. If the first qubit is 0, then the whole gate acts as the identity. But if the first qubit is 1, then the basis value of the second qubit gets flipped---the same action as the **NOT** gate $\mathbf{X}$. Hence the name Controlled-NOT, abbreviated $\mathbf{CNOT}$: the **NOT** action is controlled by the first qubit. The action on a general 2-qubit quantum state $\phi \;=\; (a,b,c,d)$ is even easier to picture:

$$\mathbf{CNOT}\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a \\ b \\ d \\ c \end{pmatrix}.$$

All it does is switch the third and fourth components---of any 4-dim. state vector. Hence, $\mathbf{CNOT}$ is a **permutation gate** and is entirely deterministic. Permuting these two indices is exactly what we need to transform the separable state $\frac{1}{\sqrt{2}}(1,0,1,0)$ into the entangled state $\frac{1}{\sqrt{2}}(1,0,0,1)$. Since we got the former state from $\mathbf{H} \otimes \mathbf{I}$ applied to $\mathbf{e}_{00}$, the matrix we want is

$$\mathbf{CNOT} \cdot (\mathbf{H} \otimes \mathbf{I})|00\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \left[\begin{array}{c|ccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{array}\right] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

We can see the result coming from the first column.