CSE491/596 Extra Notes From Chapters 5--6

These are things that were mentioned in briefer form to make the Week 13 lectures flow around Prelim II. This begins with a sizable example of the Hadamard Transform---here illustrated on 4 qubits:

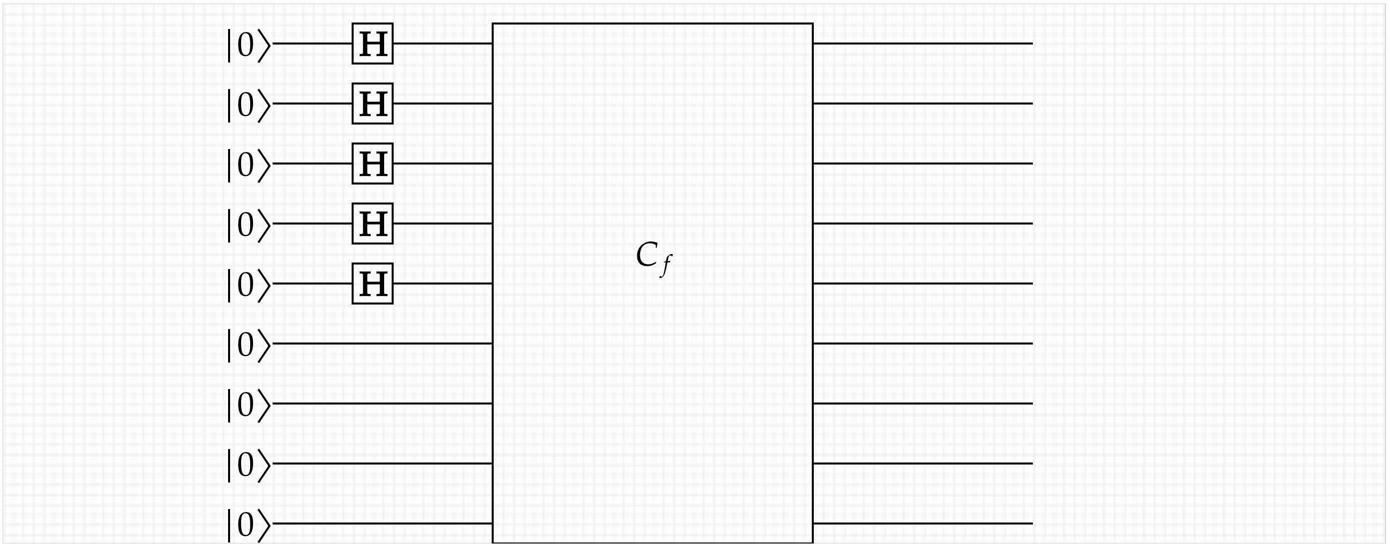
$$\left[\begin{array}{ccccccccccccccccc}
\mathbf{H} & 0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & 1000 & 1001 & 1010 & 1011 & 1100 & 1101 & 1110 & 1111 \\
0000 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0001 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
0010 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
0011 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \\
0100 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
0101 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
0110 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
0111 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1000 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1001 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1010 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\
1011 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1100 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
1101 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1110 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
1111 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1
\end{array}\right]$$

$$\mathbf{H}[u,v] \;=\; (-1)^{u \bullet v}$$

We have argued that the Hadamard transform is feasible: it is just a column of n Hadamard gates, one on each qubit line. There is, however, one consequence that can be questioned. We observed that a network of Toffoli gates suffices to simulate any Boolean circuit C (of NAND gates etc.) that computes a function $f : \{0,1\}^n \to \{0,1\}^r$. The Toffoli network C_f actually computes the reversible form

$$F(x_1, \ldots, x_n, a_1, \ldots, a_r) \;=\; (x_1, \ldots, x_n, a_1 \oplus f(x)_1, \ldots, a_r \oplus f(x)_r).$$

The matrix $\mathbf{U_f}$ of C_f is a giant permutation martrix in the 2^{n+r} underlying coordinates. Yet if the Boolean circuit C has s gates, then we reckon that C_f costs $O(s)$ to build and operate. Now build the following circuit, which is illustrated with $n = 5$ and $r = 4$:



What this circuit piece computes is the **functional superposition** of f, defined as

$$|\Phi_f\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle.$$

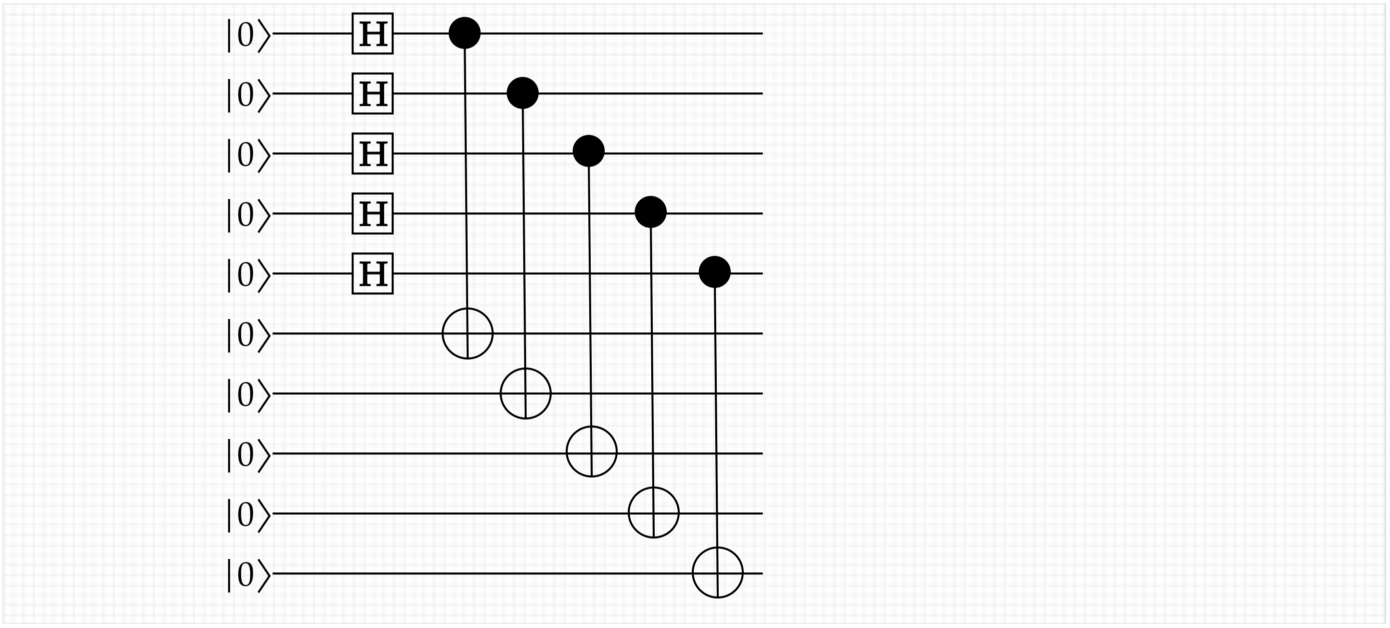
The juxtaposition of two kets really is a tensor product. This sum has exponentially many terms. It seems to preserve an exponential amount of information: the entire truth table of the Boolean function $f(x)$ over all arguments $x \in \{0,1\}^n$. However:

- f is not an arbitrary or "random" function: it is computed by a small circuit of s NAND gates.
- We cannot actually extract an exponential amount of information from $|\Phi_f\rangle$. If we measure it using the standard basis, we get our argument x back again plus r bits of some sampled function value. Measuring it in a different basis does not increase the information yield (this is part of **Holevo's Theorem**).

Nevertheless, the question remains of whether some exponential amount of "effort" must go in to the creation of $|\Phi_f\rangle$. We will "table" this question and consider the effort to be just $O(n)$ for the Hadamard transform plus $O(s)$ for the circuit.

Note About Functional Superpositions (cf. sections 6.2 and 6.4)

We've seen (on homework) that when f is the Boolean identity function on $n = 1$ bit, then C_f consists of just one CNOT gate. This generalizes for $n > 1$ using one CNOT gate per argument. Thus



computes the functional superposition

$$\frac{1}{\sqrt{32}} \sum_{x \in \{0,1\}^5} |x\rangle|x\rangle.$$

This is not the same as $|{+}{+}{+}{+}{+}\rangle \otimes |{+}{+}{+}{+}{+}\rangle$, because that is the equal superposition over all basis states for 10-bit binary strings, including all the cases of $|xy\rangle$ where the binary strings x and y of length 5 are different. An analogy is that for any set A of two or more elements, the Cartesian product of A with itself includes ordered pairs (x, y) with $x, y \in A$ but $x \neq y$, whereas the functional superposition is like the diagonal of the Cartesian product, namely $\{(x, x) : x \in A\}$. The functional superposition is entangled, just as we first saw in the case $n = 1$.

If we replace the five $\mathbf{H}$ gates by a subcircuit that prepares a general 5-qubit state

$$|\phi\rangle = a_0|00000\rangle + a_1|00001\rangle + \cdots + a_{30}|11110\rangle + a_{31}|11111\rangle,$$

then the five $\mathbf{CNOT}$ gates produce

$$D(|\phi\rangle) = a_0|0000000000\rangle + a_1|0000100001\rangle + \cdots + a_{30}|1111011110\rangle + a_{31}|1111111111\rangle.$$

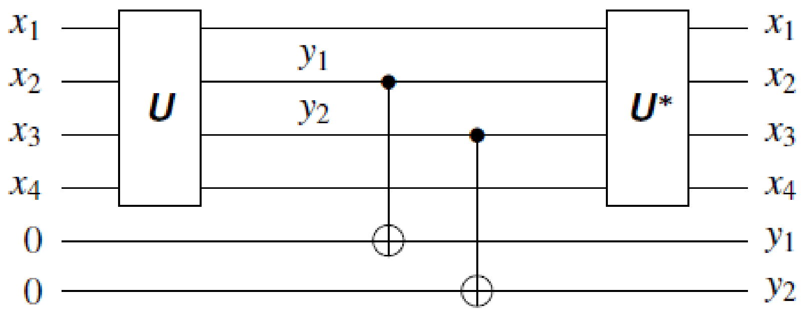
This is not the same as $|\phi\rangle \otimes |\phi\rangle$, whose terms have coefficients $a_i a_j$ for all i and j. IMHO the notation $|\phi\rangle|\phi\rangle$ or $|\phi\phi\rangle$ can be unclear about what is meant, though I've freely used $|{+}{+}\rangle$ etc. as above. When $|x\rangle$ is a basis element in the basis used for notation, then there is no difference: both $|x\rangle \otimes |x\rangle$ and $D(|x\rangle)$ have the single term $|xx\rangle$ with coefficient $1 = 1^2$.

The Copy-Uncompute Trick (section 6.3)

Now suppose we have any quantum operation $\boldsymbol{U}$ on the "x" part, where $f(x)$ might be embedded as a substring in m indexed places. We can automatically obtain the corresponding $F(x)$ via the computation

$$(\boldsymbol{U}^* \otimes \boldsymbol{I}_m)\boldsymbol{C}_m(\boldsymbol{U} \otimes \boldsymbol{I}_m)(\boldsymbol{e}_x \otimes \boldsymbol{e}_{0^m}),$$

where the $\boldsymbol{C}_m$ is applied to those index places and to m ancilla places. This effectively lifts out and copies $f(x)$ into the fresh places. The final $\boldsymbol{U}^*$ then inverts what $\boldsymbol{U}$ did in the first n places, "cleaning up" and leaving x again. Here is a diagram for $n=4$ and $m=2$ where the values $f(x)=y_1y_2 \in \{0,1\}^2$ are computed on the second and third wires and then copied to the ancillae:



This trick is called **copy-uncompute** or **compute-uncompute**. It is important to note that it works only when the quantum state after applying $\boldsymbol{U}$ and before $\boldsymbol{C}_m$ is a superposition of only those basis states that have $f(x)$ in the set of quantum coordinates to which the controls are applied. If there is any disagreement there in the superposition, then the results can be different.

On input $\boldsymbol{e}_{00}$, that is, $x_1=x_2=0$, the first Hadamard gate gives the control qubit a value that is a superposition. Hence, the second Hadamard gate does *not* "uncompute" the first Hadamard to restore $z_1=0$. The action can be worked out by the following matrix multiplication (with an initial factor of $\frac{1}{2}$):

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

This maps $\boldsymbol{e}_{00}$ to $\frac{1}{2}[1,1,1,-1]$, thus giving equal probability to getting 0 or 1 on the first qubit line.

Feasible Diagonal Matrices (sections 5.4 and 6.5)

We can continue the progression $\mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $\mathbf{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$, by

$$\mathbf{CCZ} = \begin{bmatrix} 1 &&&&&&& \\ & 1 &&&&&& \\ && 1 &&&&& \\ &&& 1 &&&& \\ &&&& 1 &&& \\ &&&&& 1 && \\ &&&&&& 1 & \\ &&&&&&& -1 \end{bmatrix}, \mathbf{CCCZ} = diag([1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,-1]),$$

and so forth. These are examples of a different kind of conversion of a Boolean function f besides the reversible form called F or C_f above. This is the matrix G_f defined for all indices u, v by

$$G_f[u,v] \;=\; \begin{cases} 0 & \text{if } u \neq v \\ -1 & \text{if } u = v \;\wedge f(u) = 1 \\ 1 & \text{if } u = v \;\wedge f(u) = 0 \end{cases}.$$

The above are G_{AND} for the n-ary AND function. The G stands for "Grover Oracle", though here I would rather emphasize that it is a concretely feasible operation.

Theorem (6.2): If f is computable by a Boolean circuit with s gates, thgen G_f can be computed by a quantum circuit of $O(s)$ gates.

When $s \;=\; s(n)$ is polynomial in n, this makes a big contrast to G_f being a 2^n-sized diagonal matrix.

The Phase Flip Trick (also section 6.5)

With reference to the idea of $a = (-1)^{f(u)}$:

This becomes a great trick if we can arrange for $\boldsymbol{a}$ itself to depend on the basis elements $\boldsymbol{e}_x$. Given a Boolean function f with one output bit, let us return to the computation of the reversible function $F(x,y) = (x, (y \oplus f(x)))$. Our quantum circuits for f have thus far initialized y to 0. Let us instead arrange $y = 1$ and then apply a single-qubit Hadamard gate. Thus, instead of starting up with $\boldsymbol{e}_x \otimes \boldsymbol{e}_0$, we have $\boldsymbol{e}_x \otimes \boldsymbol{d}$, where $\boldsymbol{d}$ is the "difference state"

$$\boldsymbol{d} = \left(\frac{1}{\sqrt{2}}, \frac{\text{-}1}{\sqrt{2}}\right) = \frac{1}{\sqrt{2}}(\boldsymbol{e}_0 - \boldsymbol{e}_1).$$

Now apply the circuit computing F. By linearity we get

$$\begin{aligned} F(x,\boldsymbol{d}) &= \frac{1}{\sqrt{2}}(F(x0) - F(x1)) \\ &= \frac{1}{\sqrt{2}}\left(\boldsymbol{e}_x \otimes \boldsymbol{e}_{0\oplus f(x)} - \boldsymbol{e}_x \otimes \boldsymbol{e}_{1\oplus f(x)}\right) \\ &= \frac{1}{\sqrt{2}}\left(\boldsymbol{e}_x \otimes (\boldsymbol{e}_{0\oplus f(x)} - \boldsymbol{e}_{1\oplus f(x)})\right) \\ &= \boldsymbol{e}_x \otimes \boldsymbol{d}', \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{d}' &= \begin{cases} \frac{1}{\sqrt{2}}(\boldsymbol{e}_0 - \boldsymbol{e}_1) & \text{if } f(x) = 0 \\ \frac{1}{\sqrt{2}}(\boldsymbol{e}_1 - \boldsymbol{e}_0) & \text{if } f(x) = 1 \end{cases} \\ &= (\text{-}1)^{f(x)}\boldsymbol{d}. \end{aligned}$$

Thus, we have flipped the last quantum coordinate by the value $a_x = (-1)^{f(x)}$. Well, actually no—by the above reasoning, what we have equally well done is that, when presented with a basis vector $\boldsymbol{e}_x$ as input, we have multiplied it by the x-dependent value a_x. We have involved the last coordinate $(n+1)$, but because we have obtained $a_x \boldsymbol{e}_x \otimes \boldsymbol{d}$, we can regard it as unchanged. In fact, we can finish with another Hadamard and ***NOT*** gate on the last coordinate to restore it to 0. On the first n qubits, over their basis vectors $\boldsymbol{e}_x$, what we have obtained is the action

$$\boldsymbol{e}_x \mapsto (-1)^{f(x)} \boldsymbol{e}_x.$$

This is the action of the Grover oracle. We have thus proved theorem 5.3 in chapter 5. We can summarize this and the conclusion of section 6.4 in one theorem statement, as follows.

THEOREM 6.2 For all (families of) functions $f\colon \{0,1\}^n \to \{0,1\}^m$ that are classically feasible, the mapping from $\boldsymbol{e}_{x0^m}$ to the functional superposition $\boldsymbol{s}_f$ and the Grover oracle of f are feasible quantum operations. □

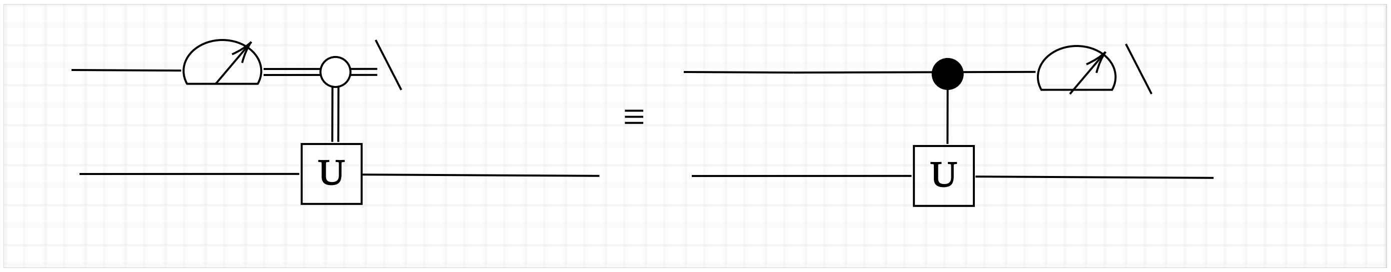
The Deferred Measurement Principle (section 6.6)

In a picture:

THEOREM 6.3 If the result b of a one-place measurement is used only as the test in one or more operations of the form "if b then ***U***," then exactly the same outputs are obtained upon replacing ***U*** by the quantum controlled operation ***CU*** with control index the same as the index place being measured and measuring that place later without using the output for control.

Proof. Suppose in the new circuit the result of the measurement is 0. Then the ***CU*** acted as the identity, so on the control index, the same measurement in the old circuit would yield 0, thus failing the test to apply ***U*** and so yielding the identity action on the remainder as well. If the new circuit measures 1, then because ***CU*** does not affect the index, the old circuit measured 1 as well, and in both cases the action of ***U*** is applied on the remainder. □

In a picture:


U
≡
U