Quantum computing: a short introduction

Alexander Shibakov

2019

A photon experiment.

...Consists of shining light through two polarizing sheets of glass with polarizations orthogonal to each other. When both sheets a placed in sequence, no light reaches the screen. When a third sheet (at 45° polarization angle) is inserted *between* the two sheets, the light shines through again.
Classical explanation is possible with the traditional light modeled as a wave (exercise). However, only the quantum mechanical explanation works for low intensity (single photon at a time) light.

Linear algebra a la Dirac: notation.

Dirac's notation for vectors, etc. is convenient in the context of tensor products (introduced later): $|v\rangle$ is the same as $\vec{v}$ in the traditional notation. To play on the connection to the polarization experiment, a simple two element basis is denoted as $|\rightarrow\rangle$ and $|\uparrow\rangle$. Now we can write linear combinations such as $|v\rangle = a|\rightarrow\rangle + b|\uparrow\rangle$. Here, a and b are usually called the *amplitudes* of $|v\rangle$ in the basis $\{|\rightarrow\rangle, |\uparrow\rangle\}$, while $|v\rangle$ is said to be a *superposition* of $|\rightarrow\rangle$ and $|\uparrow\rangle$.

A bit of physics connection.

From now on a photon=unit vector and polarizing filter=a projection on one of the basis elements.
Now, a photon $|v\rangle = a|\rightarrow\rangle + b|\uparrow\rangle$, where $|a|^2 + |b|^2 = 1$ passes through a horizontally polarized filter with *probability* (whatever this may mean) $|a|^2$ and gets absorbed with probability $|b|^2$. *After* it passes through, it becomes 'aligned' with the film. This is why placing a third polarizing sheet improves the chances of $|v\rangle$'s 'survival': $|v\rangle = c|\nwarrow\rangle + d|\nearrow\rangle$ as well!

A few details.

As a specific example, let the emitted photon be $|v\rangle = a|\rightarrow\rangle + b|\uparrow\rangle$. Suppose the first filter is polarized vertically. So $|v\rangle$ passes through the first filter with probability b^2 becoming $|\uparrow\rangle$. If the next filter is polarized horizontally then $|\uparrow\rangle$ has exactly $0^2 = 0$ chances of passing through it. On the other hand if the next filter is polarized as $|\nearrow\rangle$, since $|\uparrow\rangle = (\sqrt{2})^{-1}|\nwarrow\rangle + (\sqrt{2})^{-1}|\nearrow\rangle$, it has $((\sqrt{2})^{-1})^2 = 1/2$ probability of passing through, becoming $|\nearrow\rangle$. After that it will pass through the third (horizontally polarized filter) with probability $1/2$ (since $|\nearrow\rangle = (\sqrt{2})^{-1}|\uparrow\rangle + (\sqrt{2})^{-1}|\rightarrow\rangle$). Huh ...

A more complete picture and more notation.

We now allow a and b be *complex* numbers (to account for *circular* polarization). A quantum system (whatever that is) that can be modeled with a two dimensional (also called *two state*) space of states is called a *qubit* (short for a quantum bit). The basis of such systems is usually written as $|0\rangle$ and $|1\rangle$. All bases are assumed to be orthonormal. The inner product of $|v_1\rangle$ and $|v_2\rangle$ is written as $\langle v_1|v_2\rangle$.
A *quantum state* is an equivalence class of unit vectors $[|v\rangle]$ where $|v\rangle \sim |v\rangle'$ if $|v\rangle = e|v\rangle'$ where $|e| = 1$. Thus a qubit, mathematically, is a collection of circles in the three dimensional sphere, $\mathbb{S}^3$ (or, more precisely, a *fibration* of $\mathbb{S}^3$).

Even more notation and details. . .

Given a basis, a *ket* (or, simply, vector) can be written as a matrix $|v\rangle = a|0\rangle + b|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$. Its conjugate transpose $\langle v| = (\overline{a}, \overline{b})$ is called a *bra*. The product $\langle v_1||v_2\rangle = \langle v_1|v_2\rangle$ is usually called a *bra c ket*.

Given an n-state basis $\{ s_1, \ldots, s_n \}$ every ket $|v\rangle$ can be written as a superposition (linear combination) $|v\rangle = a_1|s_1\rangle + \cdots + a_n|s_n\rangle$.

Incidentally, $|\nwarrow\rangle$ and $|\nearrow\rangle$ form what is known as the *Hadamard* basis. We are now ready to discuss the idea of *measurement*.

Measurement I: single qubits.

A *measurement device* is simply an orthonormal basis $|0\rangle$, $|1\rangle$. The act of measurement changes a superposition $|v\rangle = a|0\rangle + b|1\rangle$ into one of the states $|0\rangle$ or $|1\rangle$ (hence, observing something always affects the outcome) with the appropriate 'probability' (a^2 or b^2). Note that a measurement does *not* measure a or b! It is more appropriate to think of measurement as a transformation. The *no cloning theorem* shown later implies that 'clever' approaches would not work either. All we can hope for is to get $|0\rangle$ or $|1\rangle$ in the end with some probability...

Quick example: quantum key exchange.

Imagine two parties (traditionally called Alice and Bob) trying to exchange a key (a sequence of classical bits) and avoid being eavesdropped on by a third party (usually called Eve, get it? ... Eve is eavesdropping ...). Here is a simple strategy:

- Alice encodes each classical bit as a qubit using a randomly chosen basis (say, either Hadamard or standard), sends it to Bob *and makes sure he recieved it*.
- Bob then randomly picks a basis for the measuring device and decodes the qubit.
- Alice sends a classical bit string telling Bob which basis was used for every bit.

Quick example: quantum key exchange, cont.

- the bases used by Bob and Alice match 50% of the time, the rest of the bits are discarded; half of the bits are used for error checking; if the error rate is too high, eavesdropping is suspected and the process is repeated.

Note that Eve cannot meddle with the quantum channel: no cloning theorem makes sure of that. This will lead to higher error rate than expected. Meddling with the classical channel does not give any insight into the contents of the message: it is simply a random stream of bits. Also note that *the timing matters*.

More notation.

The number $e^{i\phi} = \frac{a/b}{|a|/|b|}$ is called the *relative phase* of the ket $|v\rangle = a|0\rangle + b|1\rangle$. Vectors with the same relative phase and the same magnitudes of their amplitudes represent the same quantum state.

We introduce a different notation for the Hadamard basis: $|\nearrow\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|\nwarrow\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. A couple more vectors: $|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ and $|-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$.

Tensor products.

Given two vector spaces U and V, their *tensor product* is defined as the quotient space of the (infinite dimensional in general) vector space freely generated by all the pairs $|u\rangle \otimes |v\rangle \in U \times V$ with respect to the equivalence relation generated by all the identities of the form:

- $(|u_1\rangle + |u_2\rangle) \otimes |v\rangle = |u_1\rangle \otimes |v\rangle + |u_2\rangle \otimes |v\rangle$
- $|u\rangle \otimes (|v_1\rangle + |v_2\rangle) = |u\rangle \otimes |v_1\rangle + |u\rangle \otimes |v_2\rangle$
- $(a|u\rangle) \otimes |v\rangle = |u\rangle \otimes (a|v\rangle) = a(|u\rangle \otimes |v\rangle)$

Intuitively just think of adding a (somewhat deficient) 'product' operation, $\otimes$, to $+$ with all the natural 'cancelation rules'.

Tensor products, a few more details.

The tensor product $U \otimes V$ of inner product spaces has a natural inner product, defined as a linear extension of $(\langle u_1| \otimes |u_2\rangle) \cdot (\langle v_1| \otimes |v_2\rangle) = \langle u_1|v_1\rangle\langle u_2|v_2\rangle$. In addition, if $|u_1\rangle$, $\ldots$, $|u_n\rangle$ is a (n orthogonal) basis of U and $|v_1\rangle$, $\ldots$, $|v_m\rangle$ is a (n orthogonal) basis of V then $\{\, |u_i\rangle \otimes |v_j\rangle : i \leq n, j \leq m \,\}$ is a (n orthogonal) basis of $U \otimes V$. Thus $\dim U \otimes V = \dim U \times \dim V$. Note that the inner product is defined as a *linear extension* above. The reason is that most elements w of $U \otimes V$ cannot be written as $w = |u\rangle \otimes |v\rangle$ which is the mathematical manifestation of one of the mysteries of modern physics $\ldots$

Entanglement, simple facts

A state $w \in U \otimes V$ that cannot be written as $w = |u\rangle \otimes |v\rangle$ is called *entangled*. Every entangled state is a superposition of *unentangled* or *separable* states.

As an example, consided the tensor product $V_n = D \otimes D \otimes \cdots \otimes D$ of n qubit spaces. The standard basis of V_n can be chosen to consist of $|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle$, ..., $|1\rangle \otimes |1\rangle \otimes \cdots \otimes |1\rangle$. We will write $|0\rangle|0\rangle \ldots |0\rangle$ or even $|00 \ldots 1\rangle$ and so on instead, for the sake of compactness. If one recalls the binary representation of integers $0 \ldots 2^{n-1}$ then the standard basis for an n-qubit state space can be written even more compactly as $|0\rangle^n$, ..., $|2^{n-1}\rangle$ as long as the value of n is known in advance.

Entanglement, a simple example.

The basis $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, $|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$, $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$, $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ is called the *Bell basis* for a 2-qubit system.

Now $|\Phi^+\rangle$ is an entangled state, since were it so that $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle)$ then $b_1 a_2 = 0$ and $a_1 b_2 = 0$.

Also note that there are more than one way of writing a given space as a tensor product and the notion of entanglement depends on the tensor decomposition. In other words, *picking different sets of qubits will change which states are entangled*.

Measurement II: n-qubit systems.

To study multi-qubit systems we must now make our notion of measurement device more precise: a measurement device of an n-qubit system W is a direct sum decomposition $W = V_1 \oplus \cdots \oplus V_k$ (*not* a tensor decomposition).
Then given a quantum state $|w\rangle \in W$, performing a measurement produces a state (a unit vector) $|\phi\rangle \in V_i$ with 'probability' $a^2 = |P|\phi\rangle|^2$ where P is the projection onto V_i and $|w\rangle = a|\phi\rangle + |\psi\rangle$, where $|\psi\rangle \in \langle|\phi\rangle\rangle^{\perp}$.
We now show how Dirac notation makes discussing measurements (projections) more natural.

Projections, measurements, and Dirac notation

Recall the interpretation of $|v\rangle$ as a column matrix and $\langle v| = |v\rangle^*$ as its conjugate transpose. Then the product $|v\rangle\langle u|$ is a matrix that sends any vector $|w\rangle$ to $|v\rangle\langle u||w\rangle = \langle u|w\rangle|v\rangle$. If $|u\rangle$ and $|v\rangle$ are vectors in some orthonormal basis then $|u\rangle\langle v|$ sends $|v\rangle$ to $|u\rangle$ and $|u\rangle\langle u|$ is a projection on the subspace generated by $|u\rangle$.
More generally, if A is any matrix (=linear operator), we can write $(\langle u|A)|v\rangle = \langle u|(A|v\rangle) = \langle u|A|v\rangle$. Finally, $(A|v\rangle)^* = \langle v|A^*$ where A^* is the *adjoint* operator. The definition of adjoint can be made independent of the basis.

Dirac notation: example

Given the standard basis $|00\rangle, \ldots, |11\rangle$ we can now write an operator that exchanges $|00\rangle$ and $|11\rangle$ as:

$$|00\rangle\langle 11| + |11\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10|$$

and avoid using the matrix notation. It is easy to see that any linear operator may be written in Dirac notation as

$$A|v\rangle = \sum\sum a_{ij}|v_i\rangle\langle v_j|v\rangle$$

where $\{ |v_1\rangle, \ldots, |v_n\rangle \}$ is an orthogonal basis and $A(|v_j\rangle) = \sum a_{ij}|v_i\rangle$.

Hermitian formalism for direct sums.

We will now introduce a convenient (albeit somewhat redundant) way of 'coding' decompositions of W into orthogonal subspaces. Recall that an operator A is called *self-adjoint* or *Hermitian* if $A^* = A$. A vector $|v\rangle$ is called an *eigenvector* of A if $A|v\rangle = \lambda|v\rangle$ (here λ is an *eigenvalue* of A). Now, if A is self-adjoint, $\overline{\lambda}\langle v|v\rangle = (\langle v|A^*)|v\rangle = (\langle v|A)|v\rangle = \langle v|(A|v\rangle) = \langle v|\lambda|v\rangle = \lambda\langle v|v\rangle$ meaning each eigenvalue of A is real. A few more computations like this will show that any two eigenvectors corresponding to distinct eigenvalues of A are orthogonal.

Hermitian formalism for direct sums, cont.

The collection V_λ of all eigenvectors of A that correspond to the same eigenvalue λ forms a subspace of W (exercise) called the λ-*eigenspace* of A. As we have seen, all the eigenspaces of a self-adjoint operator are orthogonal. One can also show (a slightly harder exercise) that W is a direct sum of such eigenspaces. Conversely, given a direct sum $W = V_1 \oplus \cdots \oplus V_k$ one can pick a self-adjoint A with distinct eigenvalues $\lambda_1, \ldots, \lambda_k$ such that $V_i = V_{\lambda_i}$ (put $A = \sum \lambda_i P_i$ where P_i is the pojection on V_i). This gives us a convenient way of coding a direct sum decomposition of W as a self-adjoint operator. The eigenvalues do not have any significance for us, as long as they are distinct.

Hermitian formalism for direct sums, cont.

To sum it up:

- A measurement (device) M is a direct (orthogonal) sum decomposition $W = V_1 \oplus \cdots \oplus V_k$.
- The result of the measurement of a state $|v\rangle \in W$ is a unit vector $|v_i\rangle \in V_i$ with 'probability' a^2 where $|v\rangle = a|v_i\rangle + |u\rangle$, $u \in V_i^{\perp}$. If P_i is the projection operator on V_i, $|v_i\rangle = P_i|v\rangle / |P_i|v\rangle|$ and $a^2 = \langle v|P_i|v\rangle$.
- M can be represented by an *observable*: a self-adjoint operator for which each V_i is the eigenspace for some λ_i.
- The *values* of λ_i's have no significance. The *action* of M on the state is irrelevant!

More on tensor products.

If $W = U \otimes V$ and A and B are operators on U and V, respectively, we can define the operator $A \otimes B$ on W by putting $(A \otimes B)(|u\rangle \otimes |v\rangle) = A|u\rangle \otimes B|v\rangle$ and linearly extending the definition to W. If A and B are self-adjoint, then so is $A \otimes B$. Tensor products of operators are useful when dealing with measurements where one has physical access to only a portion of the system. For example in the case when the two-qubit system consists of two photons and we only have access to the first one this means that the states $a|00\rangle + b|01\rangle$ and $c|10\rangle + d|11\rangle$ must not be changed by our observation. Thus our observable will be of the form $M \otimes I$.

More on tensor products.

The tensor product of operators defined above has a simple matrix representation provided we agree on the order of the bases to be the *lexicographic* order, i.e. if the basis vector $u \in U$ is listed before $u' \in U$ and the basis vector $v \in V$ is listed before $v' \in V$ or $= v'$ then $u \otimes v$ is listed before $u' \otimes v'$.
Now if the matrix of A is $[A]$ in the basis $\{ u_1, \ldots, u_m \}$, and the matrix of B is $[B]$ in the basis $\{ v_1, \ldots, v_n \}$ the the matrix of $A \otimes B$ in the basis $\{ u_1 \otimes v_1, u_2 \otimes v_1, \ldots, u_{m-1} \otimes v_n, u_m \otimes v_n \}$ is

$$\begin{pmatrix} a_{11}B & a_{12}B & \ldots & a_{1m}B \\ & \ldots & \ldots & \\ a_{m1}B & a_{m2}B & \ldots & a_{mm}B \end{pmatrix}$$

Bell's theorem: the mathematical setup.

Consider the following two single qubit states: $|v_\theta\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle$ and $|v_\theta^\perp\rangle = -\sin\theta|0\rangle + \cos\theta|1\rangle$. Note that $|v_\theta\rangle \perp |v_\theta^\perp\rangle$ for any θ.
Let θ_1 and θ_2 be fixed and put $P_1 = (|v_{\theta_1}\rangle\langle v_{\theta_1}|) \otimes I$ and, similarly $P_2 = I \otimes (|v_{\theta_2}\rangle\langle v_{\theta_2}|)$. Recall $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. What happens when we measure $|\Phi^+\rangle$ using the projections P_1 and P_2? Applying (and keeping in mind that $I = |0\rangle\langle 0| + |1\rangle\langle 1|$)

$$P_1|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|v_{\theta_1}\rangle\langle v_{\theta_1}|0\rangle \otimes |0\rangle\langle 0|0\rangle + |v_{\theta_1}\rangle\langle v_{\theta_1}|1\rangle \otimes |1\rangle\langle 1|1\rangle)$$

Bell's theorem: the mathematical setup, cont.

Thus, after simplifying and rearranging, we get
$P_1|\Phi^+\rangle = \frac{1}{\sqrt{2}}(\cos\theta_1|v_{\theta_1}\rangle \otimes |0\rangle + \sin\theta_1|v_{\theta_1}\rangle \otimes |1\rangle) = \frac{1}{\sqrt{2}}|v_{\theta_1}\rangle \otimes |v_{\theta_1}\rangle$
with 'probability' $|P_1|\Phi^+\rangle|^2 = 1/2$. Remember, P_1 is a projection, not just a self-adjoint operator so we can find the 'probability' of the measurement having the outcome as above by just taking the square of the projection. What about P_2?
Again, $P_2(|v_{\theta_1}\rangle \otimes |v_{\theta_1}\rangle) = |v_{\theta_1}\rangle \otimes |v_{\theta_2}\rangle\langle v_{\theta_2}|v_{\theta_1}\rangle =$
$(\cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2)|v_{\theta_1}\rangle \otimes |v_{\theta_2}\rangle$ with 'probability'
$|\cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2|^2 = \cos^2(\theta_1 - \theta_2)$.

Bell's theorem: the mathemaical setup, cont.

Similarly we can compute the pobability that the consecutive measurement with P_1 and P_2 will result in both 'negative' outcomes, i.e. the result of measuring with P_1 and P_2 will be $|v_{\theta_1}^{\perp}\rangle \otimes |v_{\theta^2}^{\perp}\rangle$. Let $P_1^{\perp} = (|v_{\theta_1}^{\perp}\rangle\langle v_{\theta_1}^{\perp}|) \otimes I$ (with $P_2^{\perp}$ defined in a similar way).

Now

$P_1^{\perp}|\Phi^+\rangle = \frac{1}{\sqrt{2}}(-\sin\theta_1|v_{\theta_1}^{\perp}\rangle\otimes|0\rangle+\cos\theta_1|v_{\theta_1}^{\perp}\rangle\otimes|1\rangle) = \frac{1}{\sqrt{2}}|v_{\theta_1}^{\perp}\rangle\otimes|v_{\theta_1}^{\perp}\rangle$

so $P_2^{\perp}P_1^{\perp}|\Phi^+\rangle = \frac{1}{\sqrt{2}}((-\sin\theta_1)(-\sin\theta_2)+\cos\theta_1\cos\theta_2)|v_{\theta_1}^{\perp}\rangle\otimes|v_{\theta_2}^{\perp}\rangle$

with 'probability' $\frac{1}{2}\cos^2(\theta_1-\theta_2)$.

Bell's theorem: the physics.

We can interpret the measurements with P_1 and P_2 as the measurement of the polarization angles of a pair of photons in the state $|\Phi^+\rangle$ (the so called EPR pairs in honor of Einstein, Podolski, and Rosen). As the preceeding calculations show, the chance of both photons passing through both polarizing films or both photons being absorbed by the polarizing films is $\cos^2(\theta_1 - \theta_2)$. We now setup the following experiment: 1) a central source of pairs of photons in $|\Phi^+\rangle$ state sent in the opposite directions to Bob's and Alice's labs; 2) where each of the photons passes through a randomly and independently selected polarizing film set at one of the three angles: $0°$, $60°$, or $-60°$.

Bell's theorem: the results.

What is the result predicted by quantum mechanics? When both films are set at the same angle, *both* photons will either pass or get absorbed with probability $\cos^2 0 = 1$ (i.e. always). This happens $1/3$ of the time (since Alice and Bob choose their polarizer setting randomly and uniformly). The other $2/3$ of the time the probability of the same outcome at both Bob's and Alice's labs is $\cos^2 \phi$ where $\phi \in \{ \pm 120^\circ, \pm 60^\circ \}$ so $\cos^2 \phi = 1/4$. The total probability that both measurements are the same is $\frac{1}{3} \cdot 1 + \frac{2}{3} \cdot \frac{1}{4} = \frac{1}{2}$. This outcome is confirmed by a great many experiments.
What if the quantum mechanics is not correct? What would the result be if the photons were in a random state *right after* they left the source?

Bell's theorem: hidden variable failure.

More specifically, suppose the results are determined by some *hidden state* each of the photons has (not to be confused with the quantum state of the *pair*) and are not probabilistic at all (although the state itself is random with some distribution unknown to us). Once the state is 'set' (as soon the photons leave the source), and the film angles are decided upon (again, randomly) the outcome of the measurement of each photon is deterministic (and independent of the measurement of the other photon).
These outcomes are not entirely arbitrary, since we know from experiments that when the angles of the films are the same, the measurements always agree (i.e. both photons pass or both get absorbed).

Bell's theorem: hidden variable failure.

We will label each state according to how the photon in such state reacts to the measurement with one of the polarizer film settings: 0°, 60°, and -60°. An example of such lable is

(pass, pass, absorbed).

There are exactly 8 'classes' of states, according to this labeling scheme.
Note that both photons in the pair *must* belong to the same 'class'. Othewise, if one of them, say passes through a film set at 60° while the other gets asorbed this would violate the experimental fact that both photons react identically to the same polarizing angle setting.

Bell's theorem: hidden variable failure.

If both photons are in the state (pass, pass, pass) or (absorbed, absorbed, absorbed) the results of the measurements will be the same regardless of the angle setting. Consider the case where both photons are in the 'class' (absorbed, pass, pass). Out of 9 possible setings for the pair of polarizing films 5 ($\pm 60^\circ, \pm 60^\circ$ and $0^\circ, 0^\circ$) will produce an identical outcome (both will pass for the first four and both will get absorbed in the $0^\circ, 0^\circ$ case). This analysis holds for the remaining 5 'classes' as well. Thus in *at least* 5 out of 9 randomly chosen settings for the polarizer films, the experiment should produce an identical result in both Bob's and Alice's labs. This is at odds with the experimentally observed ratio of $1/2$.

Quantum state transformations.

A *quantum state transformation* is a unitary operator Q on the space of quantum states (so a measurement is *not* a quantum state transformation in this sense). This means that $Q^*Q = I$ so Q preserves the inner product of quantum states.
A *quantum computation* (traditionally) consists of applying a series of quantum transformations (usually picked from a small set of so called *quantum gates*) followed by one or more measurements. An alternative (and equivalent) approach to quantum computation is to *only* use measurements (which would then have to come from a larger set).

No cloning.

The security of quantum key distribution algorithm described above relied on the fact Eve did not posess a device that would allow her to measure the photon and then send the copy of the original photon to Bob. This would require a device (=linear operator) U that acts on a two-qubit space as $U|v\rangle|0\rangle = |v\rangle|v\rangle$ (i.e. *clones* $|v\rangle$). Note that any such U would be unitary, i.e. a quantum transformation (exercise).
Now if U can clone $|0\rangle$, $|1\rangle$ and $|\nearrow\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ then $U|\nearrow\rangle|0\rangle = \frac{1}{\sqrt{2}}(U|0\rangle|0\rangle + U|1\rangle|0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$ by linearity but at the same time
$U|\nearrow\rangle|0\rangle = |\nearrow\rangle|\nearrow\rangle = \frac{1}{2}(|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle)$ with an obvious contradiction. Thus such U cannot exist. The result above is known as the *no cloning principle* .

Quantum gates.

As was mentioned before, a quantum computation is simply a sequence of quantum transformations followed by a measurement. We will restrict ourselves to a finite set of transformations, each of which acts on a small number of qubits. These transformations will be called *quantum gates*. Thus the series of quantum transformations composed of quanum gates is a series of transformations of the form $I \otimes \cdots \otimes U \otimes \cdots \otimes I$.
Here are some commonly used single qubit gates:

- The four *Pauli* gates: I, $X = |1\rangle\langle 0| + |0\rangle\langle 1|$, $Y = -|1\rangle\langle 0| + |0\rangle\langle 1|$, and $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$.
- *Hadamard* gate: $H = |\nearrow\rangle\langle 0| + |\nwarrow\rangle\langle 1|$.

More quantum gates.

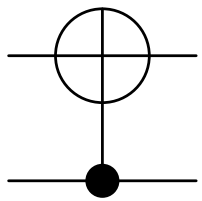
The single qubit transformations above do not affect the entanglement and, therefore, do not result in any interesting computations. The next gate, called the *controlled-NOT gate*, C_{not} is defined as $C_{not} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$. It is easy to see that C_{not} leaves $|0\rangle|0\rangle$ and $|0\rangle|1\rangle$ unchanged, sends $|1\rangle|0\rangle$ to $|1\rangle|1\rangle$ and $|1\rangle|1\rangle$ to $|1\rangle|0\rangle$, i.e. 'flips the second bit if the first one is 1'. In particular, C_{not} turns the unentangled state $|\nearrow\rangle \otimes |0\rangle$ into the entangled state $|\Phi^+\rangle$ (exercise).
A more general class of transformations, that has C_{not} as a special case is $\wedge Q = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Q$ where Q is a single qubit transformation. In this notation $C_{not} = \wedge X$.

Quantum gates: a quirk.

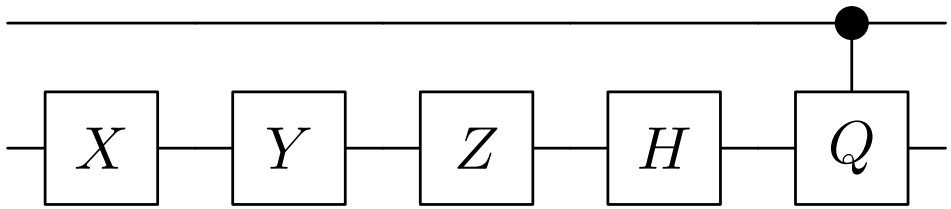
A word of caution: even when Q acts *trivially* on a (single qubit) quantum state, $\wedge Q$ might not. As an example, consider $Q|v\rangle = e^{i\phi}|v\rangle$. Note that Q does not change the quantum state, since it only affects the (physically irrelevant) global phase of the state. $\wedge Q$ however is *not* an identity transformation since it takes the state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ to $\frac{1}{\sqrt{2}}(|00\rangle + e^{i\phi}|11\rangle)$ which is *not* in the same equivalence class as the state $|\Phi^+\rangle$ (exercise). Note also that $\wedge Q$ does not change the quantum state of any of the basis vectors (since it merely multiplies them by a global phase)!

Quantum gates: notation.

As a visual aid, quantum circuits (=sequences of quantum transformations) are usually depicted schematically, with lines representing qubits. Thus, the C_{not} gate is shown as follows:



Likewise, the Pauli gates, the Hadamard gate and $\wedge Q$ would be shown as:

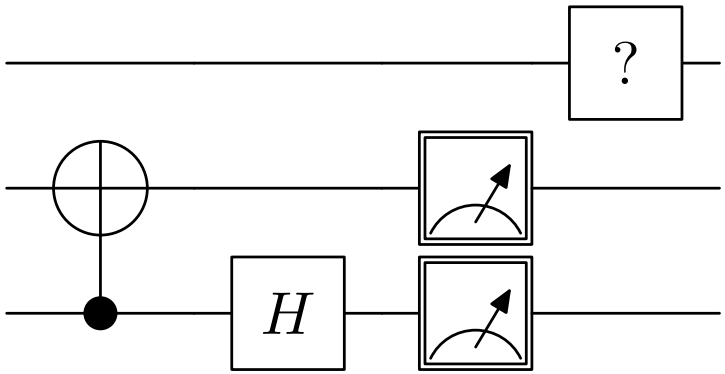


An example: quantum teleportation.

As was mentioned above, one cannot *clone* an unknown quantum state. It is therefore surprizing that one can *transmit* (or *teleport*) an unknown quantum state by using quantum transformations and a *classical* (i.e. non quantum) channel.
Let Alice have a qubit in an unknown state $a|0\rangle + b|1\rangle$. Let also Bob and Alice share an EPR pair $|\Phi^+\rangle$ (say a leftover from their Bell expeiment). Mathematically, we are dealing with a system of three qubits in the state $\frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle)$. Alice, being in posession of two of the three qubits can apply any transformation of the form $M \otimes I$ where M acts on the first two qubits.

Quantum teleportation: the circuit.

Alice proceeds to apply the following quantum circuit:



After the C_{not} is applied the state changes to $\frac{1}{\sqrt{2}}(a(|000\rangle + |011\rangle) + b(|110\rangle + |101\rangle))$. Then the Hadamard gate transforms this to $1/2(a(|000\rangle + |100\rangle + |011\rangle + |111\rangle) + b(|010\rangle - b|110\rangle + b|001\rangle - b|101\rangle))$. Factoring (in the tensor sense) the first two qubits (controlled by Alice) we can rewrite the above ...

Quantum teleportation, cont.

... as $\frac{1}{2}(|00\rangle(a|0\rangle + b|1\rangle) + |10\rangle(a|0\rangle - b|1\rangle) + |01\rangle(a|1\rangle + b|0\rangle) + |11\rangle(a|1\rangle - b|0\rangle))$. By measuring her two qubits, Alice achieves two goals: 1) the measurement itself projects the state onto one of the four possible subspaces with equal 'probability'; 2) Alice knows which subspace it is (i.e. $|00\rangle \otimes D$, or $|01\rangle \otimes D$, etc. where D is the single qubit state space, controlled by Bob). She proceeds to transmit these two classical bits (in fact, one bit of information is enough) of information to Bob.

Who, upon receiving the bits, uses one of the Pauli gates as ? to recover the quantum state (for example, if Bob receives 11 on the classical channel, he will apply Y). Bob now has the *exact* copy of Alice's original quantum state while her copy got completely destroyed by the act of measuring.

Quantum gates: generating unitary transformations.

Is a finite set of two-qubit gates sufficient to generate all quantum transformations? Obviously not, as there are uncountably many such transformations while only a countable set of them can be generated by combining finitely many gates. However, any quantum transformation can be *approximated* by a finite set of gates (in fact, one can pick the C_{not}, H and two phase rotations) to arbitrary precision.

The proof is technical and is therefore omitted. It consists of two parts: first showing that an arbitrary quantum transformation can be written as a composition of C_{not} and and a number of transformations from some well defined classes that act on a single qubit.

Quantum gates: generating unitary transformations, cont.

The second part consists of showing that the three single qubit transfromations mentioned above can approximate any single qubit transformation. As a geometric twist, it uses the fact that two rational rotations in $\mathbb{R}^3$ can generate a free group, thus their product must be an irrational rotation about some axis. (As an exercise show that $\{\sin n\}$ is dense in $[-1, 1]$).
A much harder argument shows that these approximations can be performed *efficiently* i.e. for every bit of precision (for the approximation) one has to pay only in a polynomial increase in the number of the basis gates. This is known as the *Solovay-Kitayev theorem*.

Quantum gates: generating unitary transformations, efficiency

A theorem by Knill shows that most (measure 1 in fact) unitary operators do not have an efficient *exact* representation as a product of two qubit gates.
Finally, a result by Deutsch, Barenco, and Ekert shows that most random two qubit gates can approximate *any* quantum transformation.

Quantum computation vs classical.

To estimate the relative power of a potential quantum computer we must accomplish at least two goals:

- Choose a model of computation (similar to Turing machines)
- Compare quantum computation to one of the equivalent classical computation models

Computing with quantum gates defined earlier will be our main computational model. Note that the size of a quantum algorithm built out of a fixed number of gates is also fixed. We overcome this limitation by thinking of a quantum algorithm as a (classical) procedure of picking a quantum algorithm composed of various gates for a problem of given size.

Quantum computation vs classical: Boolean circuits.

It will be implicit that this procedure is efficient (polynomial in the size of the problem) in most cases. A similar classical model of computation (i.e. a procedure of picking a boolean gate configuration for every given size problem) is equivalent to every other classical computation model (Turing, Post, Minski machines, λ-calculus, etc.).
Thus one can prove that every question about the existence of an efficient, say, Turing machine can be 'translated' into an equivalent question about the existence of an efficient (growing as a polynomial in the size of the input) Boolean circuit for a suitable Boolean function.

Boolean circuits: example.

To make matters more precise, consider an arbitrary *Boolean* function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$. We will fix a collection of Boolean functions (usually with small m and n) called *gates* and say that a sequence of $g_i \in \mathcal{B}$, $i \leq S$ is a *Boolean circuit* for f if $f(x)$ can be obtained by applying g_i's in the fixed order in sequence to the fixed 'coordinates'. We will also allow the operations that copy and ignore some coordinates of the intermediate computations.
The S above is called the *size* of the circuit.

Boolean circuits: example.

Standard examples of Boolean gates: negation, $\neg : \mathbb{F}_2 \to \mathbb{F}_2$, $\neg 0 = 1$, $\neg 1 = 0$; *exclusive or* (or equivalently, addition mod 2): $\oplus : \mathbb{F}_2^2 \to \mathbb{F}_2$ defined as $\oplus(x_1, x_2) = x_1 + x_2 \mod 2$ (or, if we remember that $\mathbb{F}_2$ is a field, $\oplus$ is the addition); (nonexclusive) or: $\vee : \mathbb{F}_2^2 \to \mathbb{F}_2$. There are many more ($\wedge$, $|$, etc.).
As an example consider a Boolean circuit for the following *addition with carry*: $x_1 +_c x_2 = (y_1, y_2)$ where $y_1 = x_1 \oplus x_2$ and $y_2 = 1$ if and only if $x_1 = x_2 = 1$. We will fix $\{ \oplus, \wedge, \neg \}$ as our gates and proceed as follows: 1) copy (x_1, x_2) to (z_1, z_2); 2) set $z_3 = z_1 \oplus z_2$; 3) set $z_4 = z_1 \wedge z_2$; 4) copy (z_3, z_4) to (y_1, y_2).

Boolean circuits: bases.

The artificial example above illustrates a few important points. The first question to ask is: how many gates are needed if we would like to be able to implement *any* Boolean function as a Boolean circuit? Such collections of gates are called *functionally complete* (we will also use the term *basis* for a functionally complete $\mathcal{B}$).
It is an amusing exercise to show that, say $\{\vee, \neg\}$ is a functionally complete set while $\{\oplus, \neg\}$ is not (the frist proof uses *normal forms* of Boolean functions while the second is an exercise in mod 2 arithmetic).
The next important point is the use of intermediate variables in the circuit above.

Boolean circuits: registers and memory.

The intermediate values represent some physical objects (wires on a chip or qubits in a quantum circuit later on). Their state at the end of the computation is *not* unimportant! We therefore amend our Boolean circuit model by requiring that all computation takes place in a fixed *memory*.

More precisely, a fixed subset $A \subseteq \{1, \ldots, N\}$ will be called a *register*. For each Boolean circuit we will fix a dedicated *input register* A and an *output register* B. Each Boolean gate will be assumed to be of the form $g : \mathbb{F}_2^k \to \mathbb{F}_2^k$ and in a Boolean circuit $g_1, \ldots, g_S$ we view each g_i as acting on some fixed register $A_i \subseteq \{1, \ldots, N\}$.

Boolean circuits: a more complete example.

To write our adder fom above in this form, first define the three gates we use as $\neg : (x_1) \mapsto (\neg x_1)$,
$\wedge : (x_1, x_2, x_3) \mapsto (x_1, x_2, x_1 \wedge x_2)$,
$\oplus : (x_1, x_2, x_3) \mapsto (x_1, x_2, x_1 \oplus x_2)$. We simply used an extra 'coordinate' to store the result of the computation.
We now set the size of our memory at 4, make $A = \{1, 2\}$ the input register and $\{3, 4\}$ the output register. The Boolean circuit works by placing some values in A (by setting x_1 and x_2), and proceeding by first applying $\oplus$ to (x_1, x_2, x_3) followed by the application of $\wedge$ to (x_1, x_2, x_4). The output appears in the output register B (i.e. (x_3, x_4)).

Boolean circuits: reversibility.

To accomplish our second goal we need to 'translate' every classical algorithm into a quantum one that performs the same computation.

We will start with some number of qubits initialized to a specific state $(q_1, \ldots, q_N)$ viewed as a classical register above. At first, it might seem we can use the Boolean model above directly, provided we can implement all the Boolean gates as quantum ones (i.e. as quantum transformations). This, however is clearly impossible, since each quantum transformation is *invertible* while, for example, the $\wedge$ gate above is clearly not (since it overwrites x_3 thus sending both $|000\rangle$ and $|001\rangle$ to $|000\rangle$). So (at a minimum) we must make all the classical computations invertible.

Boolean circuits: reversibility, first try.

We can use the following general idea to turn every Boolean function into an invertible one. Given a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$, consider instead the function $f_\tau : \mathbb{F}_2^{n+m} \to \mathbb{F}_2^{n+m}$ defined as $f_\tau(x, y) = (x, y \oplus f(x))$ (here we apply $\oplus$ to each coordinate). Now f_τ is clearly invertible (put $f_\tau^{-1}(u, v) = (u, v \ominus f(v))$) and $f_\tau(x, 0) = (x, f(x))$ so *as long as we initialize all the values outside the input register to 0* we can perform the same computations as before by using a disjoint register for output.

Boolean circuits: reversibility, problems.

The same idea can be used to make our gates invertible (thus replacing $\wedge$ with $\wedge_\tau$, etc). We can now pick a new memory location for the bits changed by each gate (so the third bit of the domain of $\wedge_\tau$ will be picked from a 'fresh' portion of the memory). It is easy to see that any Boolean function can be computed in an invertible manner.

More specifically, we can produce, for each Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$, a Boolean circuit $g_1, \ldots, g_N$ that uses invertible gates such that the Boolean circuit computes $f(x)$ for every $x \in \mathbb{F}_2^n$ in some register B of size m from the initial state $(x, 0)$.

Boolean circuits: quantum applications.

We now need the quantum analogs for a classic basis of Boolean gates. One of the Pauli gates, X can play the role of $\neg$ while, as it is easy to see $C_{not}(x_1, x_2) = (x_1, x_1 \oplus x_2)$. To create a quantum counterpart of $\wedge$ we need another gate, called the *Toffoli gate* T. Its action on on (x_1, x_2, x_3) can be described as follows:
$T = |001\rangle\langle 001| + |010\rangle\langle 010| + |011\rangle\langle 011| + |100\rangle\langle 100| + |101\rangle\langle 101| + |111\rangle\langle 110| + |110\rangle\langle 111|$, i.e. it 'flips' the last bit if and only if the first two bits are 1. A more general version of the Toffoli gate is $\wedge_2 Q$ which applies Q if and only if the first two bits are 1. Thus $T = \wedge_2 X$. Note that
$T|x_1, x_2, x_3\rangle = |x_1, x_2, x_1 \wedge x_2 \oplus x_3\rangle$ as desired.

Boolean circuits: quantum applications, cont.

The Toffoli gate, as well as $\wedge_2 Q$ can be built by combining two qubit gates exclusively (the construction will be outlined in the exercises). The C_{not} gate together with a couple of phase shifts suffice.

We can now 'reproduce' any classical computation on a quantum computer by first rewriting it as an invertible classical computation and then applying the quantum gates $\{ X, T, C_{not} \}$ (in fact, C_{not} is redundant) in place of the Boolean gates $\{\neg, \wedge_\tau, \oplus_\tau\}$. If we start with the first n qubits set to $x \in \mathbb{F}_2^n$ while the rest are set to 0 (this can be done, for example by measuring every qubit in a classical basis and then applying X to set the desired values where necessary), we can reproduce any classical algorithm.

Boolean circuits: reversibility.

Both the method of converting the classical computation into an invertible one, as well as the resulting translation of a classical computation into a quantum one have a significant flaw: they are very wasteful. Since we have no control over what the value of the 'result bit' is in each computation we have to pick a 'fresh' bit (among the ones that were set to 0 at the beginning) for the result of every gate operation. Thus the number of extra bits required to compute $f(x)$ grows at least as fast as the number of gates (size of the corresponding Boolean circuit) required to represent f. Remembering that in the quantum computer every bit is a qubit, this presents at least an implementation problem.

Boolean circuits: reversibility.

The second problem (althogh not immediately apparent) with this approach is that, in general, the intermediate bits are left entangled with the result bits in the quantum version of the algorithm. This makes it difficult if not impossible to utilize some quantum phenomena to improve the algorithm performance. 'Ignoring' the intermediate bits (by say, measuring them) as was possible in the classical case will change the global quantum state and lead to unpredictable results. More on this later.
The key to solving both problems is the following notion of a *reversible* computation.

Boolean circuits: reversibility.

Let $G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an invertible function. We say that it is represented *reversibly* by the Boolean circuit $g_1, \ldots, g_N$ acting on some memory if for every input x the Boolean circuit computes $(G(x), 0)$ from $(x, 0)$. Thus, if the register is initialized correctly, the computation will leave the intermediate values in a predictable state (0 in this case).

We will first show that *for any Boolean function* $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$, *the function* $f_\tau : (x, y) \mapsto (x, y \oplus f(x))$ *can be represented reversibly*.

As usual start with an input register A of size n, an output register B of size m and a register D, disjoint from A for the intermediate values. We can assume that $B \subseteq D \cup A$.

Boolean circuits: reversibility.

We will assume that our basis, $\mathcal{B}$ has the property that for every gate $g : \mathbb{F}_2^n \to \mathbb{F}_2^n$ in $\mathcal{B}$, there is a gate $g^{-1} \in \mathcal{B}$. Note that each of the (classical) gates T, $\neg$, and $\oplus$ are their own inverses (recall that by $\oplus$ we mean $\oplus_\tau : (x_1, x_2) \mapsto (x_1, x_1 \oplus x_2)$, what is caled C_{not} in the quantum world).
Now use our basis $\mathcal{B}$ to build a Boolean circuit $g_1, \ldots, g_N$ that computes f_τ. If (a, b, d) are the contents of the registers A, $B \cap D$ and $D \setminus B$ respectively, then the circuit $G[A, D] = g_N \circ \cdots \circ g_1$ computes $(a, f(a) \oplus b, g(a, b, d))$ where $g(a, b, d)$ represents the 'leftovers' from the computation. Let E be a register of size m disjoint from $A \cup D$.

Boolean circuits: reversibility.

By 'stringing together' m $\oplus$-gates we can create a $\tau_m[B, E]$ operation that has the effect of $\tau_m[B, E] : (b, e) \mapsto (b, e \oplus b)$ and *does not change the contents of any other registers*. Note that $\tau_m[B, E](b, 0) = (b, b)$.
Finally, by running $G[A, D]$ in reverse (i.e. by replacing each gate by its inverse) we obtain the $G[A, D]^{-1}$. Note that $G[A, D]^{-1}(a, f(a) \oplus b, g(a, b, d)) = (a, b, d)$. In the standard basis we use, $G[A, D]$ can be constructed by just applying the same gates in the reverse order.

Boolean circuits: reversibility.

The desired reversible computation is now

$$G[A, D]^{-1} \circ \tau_m[B, E] \circ G[A, D]$$

To check that it does what we need start with the value $(a, 0, 0, e)$ in the registers A, $B \cap D$, $D \setminus B$, and E, respectively. After applying $G[A, D]$ the values will change to $(a, f(a) \oplus 0, g(a, 0, 0), e) = (a, f(a), g(a, 0, 0), e)$. Then $\tau_m[B, E](a, f(a), g(a, 0, 0), e) = (a, f(a), g(a, 0, 0), e \oplus f(a))$. And the last step is $G[A, D]^{-1}(a, f(a), g(a, 0, 0), e \oplus f(a)) = (a, 0, 0, e \oplus f(a))$. Thus $A \cup E$ contains the value of $f_\tau(a, e)$.

Boolean circuits: reversibility.

Now everything is ready to show that for any *bijection* (not just the ones in the form f_τ) $G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ there is a reversible representation $g_1, \ldots, g_N$.
Let A and B be two disjoint registers of size n and let D be a register disjoint from $A \cup B$, large enough to hold the intermediate values for the *reversible* computation of G_τ and G_τ^{-1} (which is possible by the previous argument). The circuit is

$$\tau_n[A, B] \circ \tau_n[B, A] \circ G_\tau^{-1}[B, A] \circ G_\tau[A, B]$$

Starting with $(a, 0, 0)$ in A, B, and D we compute $(a, 0, 0) \mapsto (a, G(a), 0) \mapsto (0, G(a), 0) \mapsto (G(a), G(a), 0) \mapsto (G(a), 0, 0)$

Boolean circuits: benefits of reversiility.

The first observation is that by carefully counting the number of gates in our construction, we may observe that reversibility at most quadruples the number of gates representing G (not necessarily reversibly). Thus if G is effectibly computable, then it is effectibly computable *reversibly*.
The next benefit of reversibility is that it allows for a much more economical (and reversible!) computation of G by Boolean ciruits in terms of the register size. Here is a sketch of the argument.

Boolean circuits: reversibility and register size.

Let our Boolean circuit act on a register of size s and have t gates. Note that the gates do not need more that $t + s$ 'intermediate values', at least in the standard basis.
Imagine that we can split our reversible computation into two reversible 'havles' consisting of the first $t/2$ gates followed by the rest $t/2$ gates. Then the intermediate values used by the first half can be *reused* by the second, since they have all been returned to 0! In some sense, a reversible computation *uncomputes* the random intermediate results of a non reversible computation.

Boolean circuits: precise efficiency estimates.

The preious analyses can be made precise to show that for any bijection $G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ that is computable by t gates there exists a reversible Boolean circuit with $O(t^{1+\epsilon})$ gates acting on a register of size $O(n \ln t)$.

Thus we can (and will from now on) assume that every classical computation is using any required intermediate bits in a reversible way with no drop in efficiency.

Why is reversibility important in the quantum model of computation?

Boolean circuits: reversibility and quantum subrotines.

Imagine a quantum algorithm running on a system of n qubits. As is often the case, after applying a number of quantum transformations, one has to perform some classic 'postprocessing' algorithm on all the terms of the linear combination that resulted from the 'quantum' part of the algorithm (for example the classical algorithm flips some specific bits and makes it possible to cancel some of the terms in the linear combination).
This classical algorithm will need some m bits to use as intermediate values, i.e. act on a tensor product $U \otimes V$ of an n- and m-qubit spaces. If it does not return the extra qubits in the origial (0) state, its result will be (most likely) unusable, since those qubits will be left *entangled* with the original qubits. Roughly speaking, what should cancel might no longer.

Reversibility: artificial example.

Consider the following (made up) example. Suppose the result of the quantum computation is $\frac{1}{\sqrt{2}}(|110\rangle - |111\rangle)$ and the last two operations are the classical U that performs a permutation $|111\rangle \mapsto |010\rangle$, $|010\rangle \mapsto |111\rangle$ and the Hadamard transform applied to the first qubit. Suppose also that the fourth qubit is set to 0 to be used by U as an itermediate result.
If U is reversible, then $\frac{1}{\sqrt{2}}(|1100\rangle - |1110\rangle) \overset{U}{\mapsto} \frac{1}{\sqrt{2}}(|1100\rangle - |0100\rangle)$.
The Hadamard transform turns this into
$\frac{1}{\sqrt{2}}(|1100\rangle - |0100\rangle) \overset{H}{\mapsto} \frac{1}{2}(|0100\rangle - |1100\rangle - |0100\rangle - |1100\rangle) =$
$-|1100\rangle$. Now the first qubit is 1 *with certainty*!
What if U were *not* reversible?

Reversibility: artificial example, cont.

Suppose what U did was $|1110\rangle \mapsto |0101\rangle$, $|0101\rangle \mapsto |0100\rangle$, $|0100\rangle \mapsto |1110\rangle$. Note that U is still invertible and acts the same way on the first three qubits if the fourth qubit is set to 0. Now however $\frac{1}{\sqrt{2}}(|1100\rangle - |1110\rangle) \overset{U}{\mapsto} \frac{1}{\sqrt{2}}(|1100\rangle - |0101\rangle)$ and $\frac{1}{\sqrt{2}}(|1100\rangle - |0101\rangle) \overset{H}{\mapsto} \frac{1}{2}(|0100\rangle - |1100\rangle - |0101\rangle - |1101\rangle)$. Nothing cancels and the probability of the first bit being 1 is $\frac{1}{2}$! Measuring the extra bit and resetting it would not work either. While this example is artificial, the problem it exposes is not. U can be a much more complicated operator (it can, say, count the number of bits set to one), which needs extra qubits to work.

Reversibility: artificial example, cont.

The property of not leaving extra qubits entangled is not unique to the quantum analogs of the classical Boolean circuits. In a more general setting, U is a quantum transform that we would like to use as a 'subroutine' acting on n qubits. If U needs extra m bits to accomplish its computation, it should leave those extra bits *unentangled* with the first n bits.
Formally if U is an operator on $W \otimes V$ then
$U(w \otimes |0\rangle) = U'(w) \otimes a$ where a does not depend on w. That $a \neq |0\rangle$ is immaterial since it can always be measured and reset without affecting the valuable bits.

Some useful quantum transformations.

Several 'quantum subroutines' are available to design quantum algorithms. One of the most useful ones is the *Hadamard-Walsh transform*:

$$W(|q\rangle) = H(q_1) \otimes \cdots \otimes H(q_n)$$

Some useful observations:

- $W(|j\rangle) = \frac{1}{\sqrt{2}^n}(|0\rangle + (-1)^{j_1}|1\rangle) \otimes \cdots \otimes (|0\rangle + (-1)^{j_n}|1\rangle) = \frac{1}{\sqrt{2}^n} \sum_{i=0}^{2^n-1} (-1)^{i \cdot j}|i\rangle$, here $i \cdot j = \sum_{k=0}^{n} i_k j_k$, i_k is the k-th digit of i.
- $\sum_{i=0}^{2^n-1} (-1)^{i \cdot j} = \begin{cases} 2^n, & \text{if } j = 0 \\ 0, & \text{otherwise} \end{cases}$ (exercise)

Quantum oracles. Phase kickback.

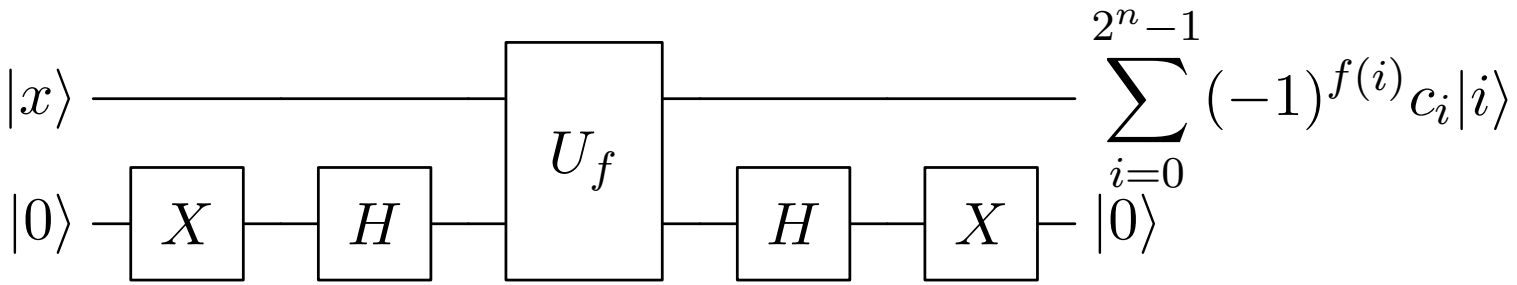
A number of quantum algorithms deal with *oracles* or *black box functions*. An oracle is simply a quantum subroutine $U_f : |x\rangle|q\rangle \mapsto |x\rangle|f(x) \oplus q\rangle$. If U_f uses any intermediate values beyond $|x\rangle$ and $|q\rangle$ it leaves them unentangled as described above. Usually U_f is a *classical* subroutine applied to a quantum system. Given a classical oracle U_f consider the following transformation: put the qubit $|q\rangle$ into the state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and apply U_f to $|x\rangle|-\rangle$. Now we compute (below $I = \{\, i : f(i) = 1 \,\}$)

Computing with a classical oracle.

$$
\begin{aligned}
U_f\left((\sum_{i=0}^{2^n-1} c_i|i\rangle)|-\rangle\right) &= \\
\frac{1}{\sqrt{2}}U_f\left((\sum_{i\in I} c_i|i\rangle + \sum_{i\notin I} c_i|i\rangle)\otimes(|0\rangle-|1\rangle)\right) &= \\
\frac{1}{\sqrt{2}}\Big(\sum_{i\in I} c_i|i\rangle\otimes|1\rangle + \sum_{i\notin I} c_i|i\rangle\otimes|0\rangle & \\
-\sum_{i\in I} c_i|i\rangle\otimes|0\rangle - \sum_{i\notin I} c_i|i\rangle\otimes|1\rangle\Big) &= \\
\frac{1}{\sqrt{2}}(-\sum_{i\in I} c_i|i\rangle + \sum_{i\notin I} c_i|i\rangle)\otimes(|0\rangle-|1\rangle) &
\end{aligned}
$$

From a classical oracle to a quantum subroutine.

Thus $U_f(|x\rangle|-\rangle) = \sum_{i=0}^{2^n-1}(-1)^{f(i)}c_i|i\rangle \otimes |-\rangle$, i.e. this transformation negates all the terms corresponding to i's such that $f(i) = 1$. We can put it together in the following quantum circuit:



The circuit above is a true quantum subroutine in that it does not entangle the extra bits with the bits in the output (returning the intermediate qubit back to $|0\rangle$ is not strictly necessary). We will use the notation I_{U_f} for the circuit above. Next we shall use it in our first quantum algorithm.

Deutsch-Jozsa problem: the setup.

Suppose a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is known to be one of the following two kinds: f is either *constant* or *balanced*, i.e. f takes on the value of 1 exactly the same number of times it takes on the value of 0. The *Deutsch-Jozsa problem* consists of designing a quantum algorithm that given a (classical) oracle U_f that computes f decides whether f is constant or balanced.
It is easy to see that any classical algorithm would have to call f at least 2^{n-1} times to make this determination. Below we will build a quantum algorithm that can accomplish the same task after calling U_f *once*!

Deutsch-Jozsa problem: the superposition.

The algorithm starts by preparing a quantum state that is a superposition of all the possible inputs of f. This can be achieved in the following manner. Start with a state of $|0\rangle^n$ (which can be initialized by measuring each qubit of an arbitrary quantum state $|x\rangle$ in the standard basis and then applying one of the Pauli gates X to 'invert' the qubits that are in the wrong state). Then apply the Walsh-Hadamard transformation

$$W(|0\rangle^n) = \frac{1}{\sqrt{2}^n} \sum_{i=0}^{2^n-1} (-1)^{i\cdot 0} |i\rangle = \frac{1}{\sqrt{2}^n} \sum_{i=0}^{2^n-1} |i\rangle$$

Deutsch-Jozsa problem: applying the oracle.

We now apply the circuit I_{U_f} built above to the superposition just constructed (after introducing an auxilary qubit initialized to $|0\rangle$):

$$I_{U_f}\left(\frac{1}{\sqrt{2}^n}\sum_{i=0}^{2^n-1}|i\rangle\right)=\frac{1}{\sqrt{2}^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)}|i\rangle$$

followed by another Walsh-Hadamard transformation:

$$W(\frac{1}{\sqrt{2}^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)}|i\rangle)=\frac{1}{2^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)}\sum_{j=0}^{2^n-1}(-1)^{i\cdot j}|j\rangle$$

All that is left to do is to measure the resulting state in the standard basis.

Deutsch-Jozsa problem: the result.

There are two possible cases. If f is constant then

$$\begin{aligned}
\frac{1}{2^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)}\sum_{j=0}^{2^n-1}(-1)^{i\cdot j}|j\rangle &= \\
\frac{(-1)^{f(0)}}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}(-1)^{i\cdot j}|j\rangle &= \\
\frac{(-1)^{f(0)}}{2^n}\sum_{j=0}^{2^n-1}\left(\sum_{i=0}^{2^n-1}(-1)^{i\cdot j}\right)|j\rangle &= (-1)^{f(0)}|0\rangle^n
\end{aligned}$$

Thus our measurement will return a 0 state *with certainty*!

Deutsch-Jozsa problem: the setup.

If, on the other hand, f is *balanced* then

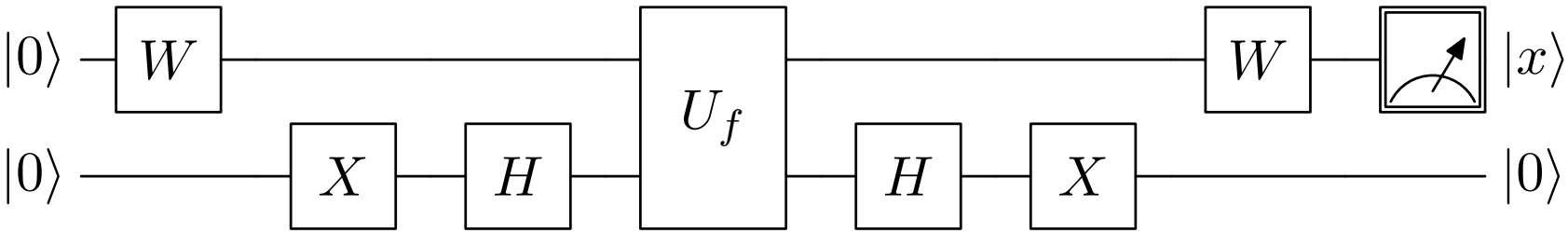
$$\frac{1}{2^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)}\sum_{j=0}^{2^n-1}(-1)^{i\cdot j}|j\rangle \quad =$$

$$\frac{1}{2^n}\sum_{j=0}^{2^n-1}\left(\sum_{f(i)=0}(-1)^{i\cdot j}-\sum_{f(i)=1}(-1)^{i\cdot j}\right)|j\rangle$$

If $j=0$ the difference inside the parentheses above is also 0 since f is balanced. Thus our measurement will return $\neq |0\rangle^n$ *with certainty*! Let us look at the corresponding quantum circuit.

Deutsch-Jozsa problem: the circuit.

In the circuit below it is very apparent that the oracle was only applied once. The quantum nature of the algorithm allowed us to use the oracle essentially *on every input at the same time*!



The effect exploited above is given the name of *quantum parallelism*, i.e. applying a (usually classical) quantum subroutine to a superposition. It is somewhat abused in popular explanations of quantum algorithms.

Simon's problem: stochastic quantum computation.

The algorithms considered so far produce results *with certainty*. We now consider a problem where the computation is probabilistic. In *Simon's problem* $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is a 2-1 function such that for some bit string a and every x the value $f(x) = f(x \oplus a)$. The goal is to compute a. Here, f is a 'black-box' function and the task is to minimize the number of applications of f to find a. The best classical algorithm can solve this problem in $O(2^{n/2})$ steps. Simon's *quantum* algorithm can solve the problem in $O(n)$ calls to U_f with $O(n^3)$ postprocessing steps.

Simon's problem: stochastic quantum computation.

Start by creating a superposition $\left(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^{n-1}}|x\rangle\right)\otimes|0\rangle^n$ using a Walsh-Hadamard transform. Apply U_f to produce $\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^{n-1}}|x\rangle|f(x)\rangle$. Now *measure the right register in the standard basis*.

The measurement result itself is ignored. Recall that such a measurement will project the superposition above on some $\mathbb{F}_2^n\otimes|y\rangle$. After the measurement, the input (left) register will contain $\frac{1}{\sqrt{2}}(|x_0\rangle+|x_0\oplus a\rangle)$ where the measured value was $f(x_0)$.

Simon's problem: stochastic quantum computation.

Proceed to apply the Walsh-Hadamard transform to the result:

$$W(\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle)) = \frac{1}{\sqrt{2^{n+1}}} \sum_{i=0}^{2^{n-1}} ((-1)^{x_0 \cdot i} + (-1)^{(x_0 \oplus a) \cdot i})|i\rangle.$$

Now $(x_0 \oplus a) \cdot i = x_0 \cdot i \oplus a \cdot i$ so the last sum is

$$\frac{1}{\sqrt{2^{n-1}}} \sum_{i \in S} (-1)^{x_0 \cdot i} |i\rangle$$

where for each $i \in S$,

$$x_0 \cdot i = x_0 \cdot i \oplus a \cdot i \mod 2,$$

Thus $a \cdot i = 0 \mod 2$ (remember $\oplus$ is addition in $\mathbb{F}_2$)!

Simon's problem: stochastic quantum computation.

Rewrite $a \cdot i = 0 \mod 2$ as $i_1 \cdot a_1 \oplus \cdots \oplus i_n \cdot a_n = 0$ to see that this is a *linear equation in* $a_1, \ldots, a_n$ *over* $\mathbb{F}_2$. Now measuring the *left* register will give us a random (and uniformly distributed) equation of this kind.

The next step is to repeat the measurement $n-1$ times. If the 'binary vectors' $|i^k\rangle$ are linearly independent for $k = 1, \ldots, n-1$ then we can solve the (underdetermined, since $a \neq 0$) system of linear equations thus obtained in $O(n^3)$ steps (by Gaussian elimination).

Simon's problem: stochastic quantum computation.

What is the probability that the equations are linearly independent? The pobability of i^{k+1} being independent of $i^1, \ldots, i^k$ is $\dfrac{\text{size of } \mathbb{F}_2^n - \text{size of } \mathbb{F}_2^k}{\text{size of } \mathbb{F}_2^n} = \dfrac{2^n - 2^k}{2^n} = 1 - \dfrac{1}{2^{n-k}}$.
Picking i^k independently at random the probability is

$$(1 - \frac{1}{2^{n-1}}) \cdots (1 - \frac{1}{2}) \geq \prod_{k=1}^{\infty} (1 - \frac{1}{2^k})$$

Using $(1-a)(1-b) \geq 1-(a+b)$ for positive a and b, the product above is at least $\dfrac{1}{4}$ (start by $\prod_{k=1}^{\infty}(1 - \frac{1}{2^k}) = \frac{1}{2}\prod_{k=2}^{\infty}(1 - \frac{1}{2^k})$).

Simon's problem: stochastic quantum computation.

Now one can repeat the computation above to reduce the probability of error exponentially to an arbitrary small value.
It may be of some interest to point out that there is an *exact* algorithm that solves Simon's problem (i.e. it is guaranteed to produce a result in polynomial time) but it is much more involved than the construction above.
Another remark of interest, although not related to quantum computation, is that the product $\prod_{k=1}^{\infty}(1-\frac{1}{2^k})$ can be estimated much more precisely using the *pentagonal number theorem* by L. Euler.

Quantum Fourier Transform, preliminaries: DFT and FFT.

Recall the definition of the Discrete Fourier Transform (DFT). Given a polynomial $P_n(z) = c_0 + c_1 z + \cdots + c_n z^{n-1}$ and the *principal n-th root of unity* $\omega_n = e^{\frac{2\pi i}{n}}$ the DFT of P_n is the sequence $\check{P}_n = P_n(\omega_n^0), \ldots, P_n(\omega_n^{n-1})$. The straightforward evaluation of $\check{P}_n$ takes $O(n^2)$ steps (even that requires a minor trick to evaluate the value of $P_n(z)$ in $O(n)$ steps).
The *Fast Fourier Transform* is an algorithm to evaluate the DFT in $O(n \ln n)$ steps. It is based on the following ideas.

The Fast Fourier transform.

For simplicity, assume below that n is a power of 2 (this is a simplifying idea 0, if you will).

- $P_n(z) = P^e_{n/2}(z^2) + zP^o_{n/2}(z^2)$ where $P^e_{n/2}(z) = c_0 + c_2 z + \cdots + c_{n-2} z^{n/2-1}$ and $P^e_{n/2}(z) = c_1 + c_3 z + \cdots + c_{n-1} z^{n/2-1}$ (i.e. the even and the odd parts of P_n).
- $\omega_n^2 = e^{\frac{2\cdot 2\pi i}{n}} = e^{\frac{2\pi i}{n/2}} = \omega_{n/2}$.

Putting the two together we get

- $\check{P}^k_n = P^e_n(\omega_n^{2k}) + \omega_n^k P^o_n(\omega_n^{2k}) = (\check{P}^e_{n/2})^k + \omega_n^k (\check{P}^o_{n/2})^k$

Fast Fourier Transform: efficiency.

Using the last formula it is easy to see that it takes $T(n) = 2T(n/2) + O(n)$ steps to compute the transform. Here $2T(n/2)$ is the 'time' (measured as the number of elementary operations) to compute the two transforms of size $n/2$ and $O(n)$ is the time to perform the summation and multiplication for the n values of $\check{P}_n$. Now $T(n) = O(n \ln n)$ by a well known result in combinatorics (just plot the recursive call tree for the algorithm to see this). Note that $O(n)$ also includes the steps necessary to 'reshuffle' the coefficients of P_n.

One can also provide a speedup (not asymptotically but still significant) by noticing that $\omega_n^{k+\frac{n}{2}} = e^{\frac{(k+n/2)2\pi i}{n}} = -\omega_n^k$.

The matrix representation of FFT.

We now rewrite the FFT formula above in the matrix form. First, some notation. We will assume that the length of the sequence is 2^n. Since FFT is a linear transform it can be represented by a $2^n \times 2^n$ matrix F^n. We will use the notation S^n for the 'shuffle' matrix that sends the basis vector $|2i\rangle$ to $|i\rangle$ and the vector $|2i+1\rangle$ to $|\frac{n}{2}+i\rangle$ (note we are using Dirac's notation for convenience only at this time). That is, S^n splits the polynomial into its even and odd halves. Note that the *result* of S^n can be achieved in $O(n)$ *bit permutations* while S^n performs 2^n *write operations*.
Finally, the matrix D^n is the diagonal $2^n \times 2^n$ matrix with the entries $\omega_{2^{n+1}}^0, \ldots, \omega_{2^{n+1}}^{2^n-1}$ on the diagonal.

The matrix representation of FFT.

The formula for the FFT becomes:

$$F^n = \begin{pmatrix} I^{n-1} & D^{n-1} \\ I^{n-1} & -D^{n-1} \end{pmatrix} \begin{pmatrix} F^{n-1} & 0 \\ 0 & F^{n-1} \end{pmatrix} S^n$$

We now define the *Quantum Fourier Transform* as a quantum transformation

$$U_{F^n} : |i\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum \omega_{2^n}^{ij} |j\rangle$$

The constant $\frac{1}{\sqrt{2^n}}$ is necessary to make U_{F^n} unitary (it is missing in our formula for FFT for simplicity).

From the matrix form of FFT to the efficient QFT.

We can use the formula for the classical FFT to represent U_{F^n} as

$$U_{F^n} = \frac{1}{\sqrt{2}} \begin{pmatrix} I^{n-1} & D^{n-1} \\ I^{n-1} & -D^{n-1} \end{pmatrix} \begin{pmatrix} U_{F^{n-1}} & 0 \\ 0 & U_{F^{n-1}} \end{pmatrix} S^n$$

in the standard basis. Note that $\begin{pmatrix} U_{F^{n-1}} & 0 \\ 0 & U_{F^{n-1}} \end{pmatrix}$ is simply $I \otimes U_{F^{n-1}}$, i.e. the QFT on the lower $n-1$ qubits. We will now show that $D^k = D^{k-1} \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_{2^{k+1}} \end{pmatrix}$

QFT: the implementation.

To see the last formula, let $|i\rangle$ be a vector in the standard basis. Suppose j is even so $|j\rangle$ has the form $|j/2\,0\rangle = |j/2\rangle \otimes |0\rangle$. Then

$$\begin{aligned} D^{k-1} \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_{2^{k+1}} \end{pmatrix} |j/2\rangle \otimes |0\rangle &= \\ \omega_{2^k}^{j/2} |j/2\rangle \otimes |0\rangle &= \omega_{2^{k+1}}^{j} |j\rangle = D^k |j\rangle \end{aligned}$$

Now consider the case of an odd j.

QFT: the implementation, cont.

Similarly if $|j\rangle = |(j-1)/2\rangle \otimes |1\rangle$ then

$$\begin{aligned} D^{k-1} \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_{2^{k+1}} \end{pmatrix} |(j-1)/2\rangle \otimes |1\rangle &= \\ \omega_{2^k}^{\frac{(j-1)}{2}} |(j-1)/2\rangle \otimes (\omega_{2^{k+1}}|1\rangle) &= \\ \omega_{2^{k+1}}^{j-1}\omega_{2^{k+1}}|(j-1)/2\rangle \otimes |1\rangle &= D^k|j\rangle \end{aligned}$$

Armed with this recursive relation we can now compute

$$\frac{1}{\sqrt{2}} \begin{pmatrix} I^{n-1} & D^{n-1} \\ I^{n-1} & -D^{n-1} \end{pmatrix} |\delta\rangle \otimes |j\rangle, \quad \delta = 0, 1$$

QFT: the implementation, cont.

First consider the case $\delta = 0$ (remember, δ is the *most significant bit*). Then

$$\frac{1}{\sqrt{2}} \begin{pmatrix} I^{n-1} & D^{n-1} \\ I^{n-1} & -D^{n-1} \end{pmatrix} |0\rangle \otimes |j\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |j\rangle + |1\rangle \otimes |j\rangle)$$

If $\delta = 1$ we have

$$\frac{1}{\sqrt{2}} \begin{pmatrix} I^{n-1} & D^{n-1} \\ I^{n-1} & -D^{n-1} \end{pmatrix} |1\rangle \otimes |j\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes D^{n-1}|j\rangle - |1\rangle \otimes D^{n-1}|j\rangle)$$

QFT: the implementation, cont.

After factoring we get

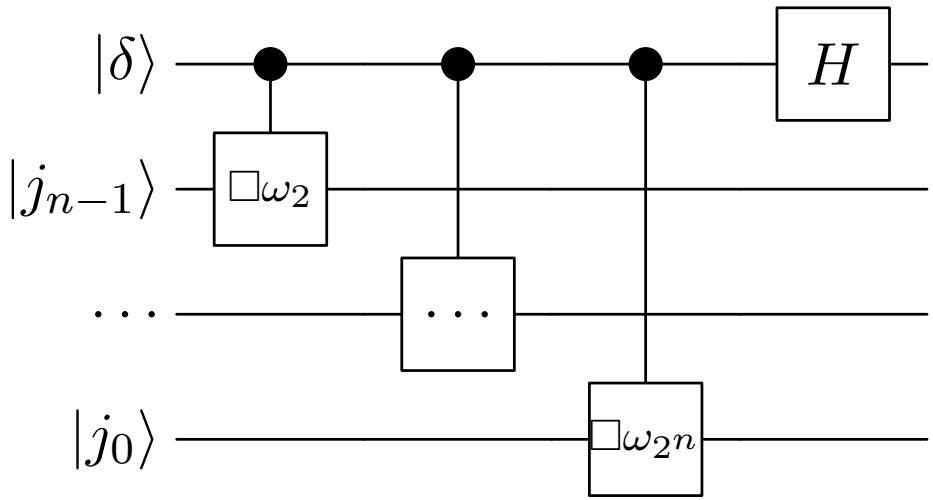
$$\frac{1}{\sqrt{2}}\begin{pmatrix} I^{n-1} & D^{n-1} \\ I^{n-1} & -D^{n-1} \end{pmatrix}|\delta\rangle \otimes |j\rangle = H \otimes I^{n-1}(|0\rangle\langle 0| \otimes I^{n-1} + |1\rangle\langle 1| \otimes D^{n-1})|\delta\rangle|j\rangle$$

The operator in parentheses, $(|0\rangle\langle 0| \otimes I^{n-1} + |1\rangle\langle 1| \otimes D^{n-1})$ is simply D^{n-1} *controlled* by δ. Since

$$D^{n-1} = D^{n-2} \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_{2^n} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \omega_2 \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_{2^n} \end{pmatrix}$$

QFT: the implementation, cont.

we can use the following quantum circuit to compute the last stage.



where $\square\omega_{2^k} = \begin{pmatrix} 1 & 0 \\ 0 & \omega_{2^k} \end{pmatrix}$ so this circuit takes $O(n)$ gates to implement.

QFT: the efficiency.

Reviewing the QFT circuit

$$U_{F^n} = \frac{1}{\sqrt{2}} \begin{pmatrix} I^{n-1} & D^{n-1} \\ I^{n-1} & -D^{n-1} \end{pmatrix} \begin{pmatrix} U_{F^{n-1}} & 0 \\ 0 & U_{F^{n-1}} \end{pmatrix} S^n$$

we note that S^n takes up n (2-qubit permutation) gates, and that the leftmost operator takes up another n gates. The middle operator *is just the QFT on* $n-1$ *qubits* so recursively expanding the circuit we obtain the QFT efficiency to be $O(n^2)$ compared to the classical version for which the efficiency is $O(n2^n)$ in our notation!

QFT: the efficiency, cont.

To compare the classical version with the quantum one note that in the recursive relation, the classical FFT requires *two transforms* each performed on *half the samples* while the post- and preprocessing steps require $O(2^n)$ operations. Thus the recurrent relation for the number of steps is $T(2^n) = 2T(2^{n-2}) + O(2^n)$ giving $T(2^n) = O(n2^n)$. The QFT, on the other hand, requires just *one transform* applied to *qubits less one* and the post- and preprocessing stages take up $O(n)$ gates so the relation is $T(n) = T(n-1) + O(n)$ resulting in $T(n) = O(n^2)$.
Also note that the two transforms (FFT and QFT) are not quite the same: whereas the FFT produces a sequence of individually accessible values, the QFT turns one quantum state into another.

QFT: applications, Shor's algorithm.

The QFT forms the core of P. Shor's efficient (probabilistically polynomial, i.e. better than $O(n^3)$) algorithm for factoring large numbers (the meaning of n will become clear soon).
Suppose a large number M is given by its sequence of m bits. To find a nontrivial factor of M one can try the following method;

- Guess a random $a < M$ such that $\gcd(a, M) = 1$ (if not, we are done);
- Find an *even* r such that $a^r = 1 \mod M$ (r is called the *period* of a (half of a's will have such r);
- $(a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) = 0 \mod M$ so probably $(a^{\frac{r}{2}} \pm 1)$ has a common factor with M; use Euclidean algorithm to find it;

QFT: applications, Shor's algorithm, cont.

The first and last step are efficient in the classical sense. The second step is the one for which no efficient classical algorithm is known.

Shor's idea was to use the QFT to guess the period for a. He starts by picking an n such that $M^2 \leq 2^n < 2M^2$ (so $n \sim 2\log_2 M$). The function $f(x) = a^x \mod M$ is efficiently computable classically (for each x) so there exists an efficient quantum circuit for U_f (recall that $U_f : |x\rangle|0\rangle^m \mapsto |x\rangle|f(x)\rangle$).

QFT: applications, Shor's algorithm, cont.

Start the computation by creating a superposition

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |0\rangle^m \overset{U_f}{\mapsto} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle$$

using Walsh-Hadamard transform followed by U_f.
Now measure the last m bits (i.e. a random value of f). The measurement will project the state to one of

$$C \sum_x g(x)|x\rangle |f_0\rangle$$

where $g(x) = 1$ if and only if $f(x) = f_0$.

QFT: applications, Shor's algorithm, cont.

Discard f_0 and observe that if $g(x) = g(x') = 1$ for some $x \neq x'$ then $x - x'$ is a multiple of the period of f. So f and g *have the same period*. To find it, apply the QFT:

$$F^n(C\sum_x g(x)|x\rangle) = C\sum_x \check{g}(x)|x\rangle$$

and observe, that like in the case of FFT $\check{g}(x)$ will concentrate around x that are multiples of the period of g. Measure one. The quantum part is over. Shor shows how to turn the guess thus obtained into a value that is the period most of the time.

Concluding remarks.

To complete the analysis of Shor's algorithm one would have to show that all the probabilities are appropriately bounded and that all the classical portions are efficient. We omit such details as they are fun exercises to do.

Note that it is not known whether there is a *classical* algorithm for efficient factoring. It is widely believed that no such algorithm exists and most cryptographic systems are based on this belief.

It is currenly not known whether there is an algorithm like Shor's that has a certain outcome.

Finally, there *is* a problem that can be solved more efficiently using quantum computing (Grover's algorithm) but *not* with an exponential speedup.