

Generating all unitaries

In the quantum circuit model, all computations are done by applying a succession of *quantum gates*. These are unitary transformations acting on one, two or three q-bits at a time. By applying them successively, one builds up a unitary transformation on the full 2^n -dimensional Hilbert space of n q-bits.

For classical circuits, there are *universal sets* of gates which can be used to build up any Boolean function on n bits. Examples of such sets are: NOT, AND, and OR (or just NOT and AND or NOT and OR), together with crossover and fan-out; the NAND gate, together with crossover, fan-out and ancillas; or for reversible circuits, the Toffoli gate, together with ancillas and crossover. Is there a quantum analog of such universal sets, which enable one to build up any unitary transformation on n q-bits?

Single-bit gates

As we have seen, the most general unitary on a single q-bit can be written

$$e^{i\alpha} \exp(i\theta(\vec{n} \cdot \hat{\vec{\sigma}})/2),$$

where \vec{n} is a real unit 3-vector (whose direction can be parametrized by two angles η and ϕ) and $\hat{\vec{\sigma}}$ is $(\hat{X}, \hat{Y}, \hat{Z})$. The overall phase α is physically irrelevant, so we will neglect it henceforth.

Each of these single-bit unitaries can be considered a quantum gate. Therefore, there is a three-parameter family of single-bit gates, parametrized by θ, η, ϕ . No set consisting solely of one-bit gates, however, can be universal. This is easy to see by noting that no combination of such gates can ever produce entanglement.

$$\begin{aligned} (\hat{U}_1 \otimes \cdots \otimes \hat{U}_n)(|\phi_1\rangle \otimes \cdots \otimes |\phi_n\rangle) = \\ (\hat{U}_1|\phi_1\rangle) \otimes \cdots \otimes (\hat{U}_n|\phi_n\rangle). \end{aligned}$$

In fact, we can reduce our three-parameter family of single-bit gates to two one-parameter families. We have already seen that (up to a phase) all one-bit unitaries are isomorphic to rotations in three dimensions. We can define the *rotation operator*

$$\hat{R}_{\vec{n}}(\theta) \equiv \exp(-i(\theta/2)(\vec{n} \cdot \hat{\vec{\sigma}})).$$

We can therefore draw upon a well-known theorem from classical geometry that any rotation in three dimensions about an arbitrary axis can be performed by doing three rotations about two *fixed* axes. If we take our axes to be X and Z , this means

$$\hat{R}_{\vec{n}}(\theta) = \hat{R}_Z(\gamma)\hat{R}_X(\beta)\hat{R}_Z(\alpha)$$

for some angles α, β, γ .

(In fact, the two fixed axes don't have to be at right angles, though if they are not it may take more than three rotations to generate any rotation.)

Single-bit gates and the CNOT

Unlike single-bit gates, most two-bit gates can produce entanglement. The canonical example of this is the CNOT.

$$\hat{U}_{\text{CNOT}}(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle = \alpha|00\rangle + \beta|11\rangle.$$

We can show that any unitary transformation on n q-bits can be built up from CNOTs between pairs of bits and the infinite family of single-bit unitary gates.

To show this, however, we must first prove a number of intermediate results.

Lemma 1. Any $N \times N$ unitary matrix \hat{U} can be built up as a product of unitary matrices \hat{V}_i of the form

$$\hat{V} = \begin{pmatrix} 1 & 0 & \cdots & & \cdots & 0 \\ 0 & 1 & \cdots & & \cdots & 0 \\ \vdots & \vdots & \ddots & & & \vdots \\ & & & a & b & \\ & & & c & d & \\ \vdots & \vdots & & & & \vdots \\ 0 & 0 & \cdots & & \cdots & 1 \end{pmatrix},$$

which are the identity except on one 2×2 submatrix.

We call such matrices *two-level unitaries*.

Proof. First note that for any 2-vector (x, y) , there is a 2×2 unitary such that

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \sqrt{|x|^2 + |y|^2} \\ 0 \end{pmatrix}.$$

The solution is

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \frac{1}{\sqrt{|x|^2 + |y|^2}} \begin{pmatrix} x^* & y^* \\ -y & x \end{pmatrix},$$

which satisfies the previous equation and is manifestly unitary.

It therefore follows that for an N -vector (ξ_1, \dots, ξ_N) , there is a succession of $N - 1$ two-level unitary matrices $\hat{V}_1, \dots, \hat{V}_{N-1}$ of size $N \times N$ such that

$$\hat{V}_{N-1} \cdots \hat{V}_1 \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_N \end{pmatrix} = \begin{pmatrix} \sqrt{\langle \xi | \xi \rangle} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

One chooses these matrices \hat{V}_i to successively set the elements of the vector to 0, starting at the bottom and working up to the top.

Now let us consider an $N \times N$ unitary matrix \hat{U} . Its inverse is

$$\hat{U}^\dagger = \begin{pmatrix} \xi_1 & \xi'_2 & \cdots & \xi'_N \\ \xi_2 & \ddots & & \\ \vdots & & & \\ \xi_N & \cdots & & \end{pmatrix}$$

Let us now find unitaries $\hat{V}_1, \dots, \hat{V}_{N-1}$ to convert the first column of \hat{U}^\dagger to $(1, 0, \dots, 0)$. Because $\hat{V}_{N-1} \cdots \hat{V}_1 \hat{U}^\dagger$ is still unitary, all the columns must remain orthogonal; so the first row must be $(1, 0, \dots, 0)$ as well. The new matrix is

$$\hat{V}_{N-1} \cdots \hat{V}_1 \hat{U}^\dagger = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & (\hat{U}')^\dagger & \\ 0 & & & \end{pmatrix},$$

where \hat{U}' is an $(N - 1) \times (N - 1)$ unitary matrix.

We can then repeat the same trick on the $(N - 1) \times (N - 1)$ submatrix by finding some $N - 2$ two-level unitaries to leave us with a new matrix of the form

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & & & \\ \vdots & \vdots & & (\hat{U}'')^\dagger & \\ 0 & 0 & & & \end{pmatrix},$$

where \hat{U}'' is an $(N - 2) \times (N - 2)$ unitary; and so on. In the end, we will have found $N(N - 1)/2$ unitary matrices \hat{V}_i such that

$$\hat{V}_{N(N-1)/2} \cdots \hat{V}_1 \hat{U}^\dagger = \hat{I},$$

which implies that

$$\hat{V}_{N(N-1)/2} \cdots \hat{V}_1 = \hat{U}.$$

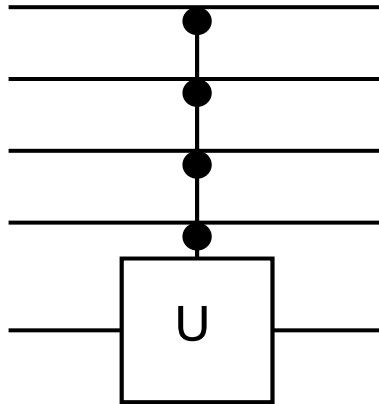
We now show that any unitary of type \hat{V} can be realized by CNOTs and single-bit gates. The important insight is that any 2×2 unitary is equivalent to a single-bit gate on the subspace spanned by those two vectors.

Assume that $N = 2^n$, i.e., this is a space of n q-bits. A gate on one q-bit $\hat{I} \otimes \hat{I} \otimes \dots \otimes \hat{G}$ does *not* have the 2-level form; instead it looks like

$$\begin{pmatrix} \hat{G} & 0 & \dots & 0 \\ 0 & \hat{G} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & & \hat{G} \end{pmatrix}.$$

To build a two-level unitary, we instead need a controlled-controlled- \dots -controlled gate.

That is, a circuit of the form



gives us a unitary matrix of the form

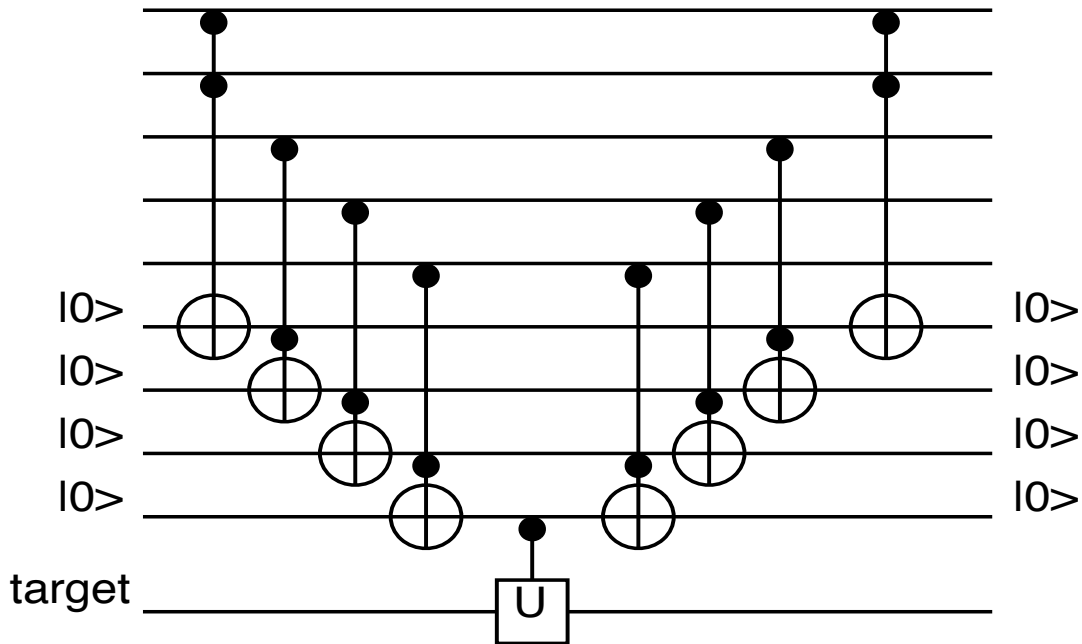
$$\hat{V} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & & \vdots \\ \vdots & \vdots & \ddots & & \\ & & & a & b \\ 0 & \cdots & & c & d \end{pmatrix},$$

where

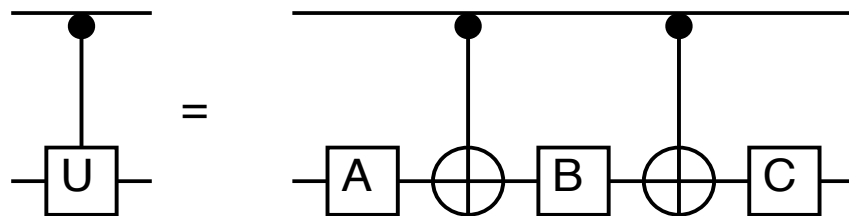
$$\hat{U} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

We can build this kind of gate out of Toffoli and controlled-U gates using ancillas.

E.g., for $n = 5$

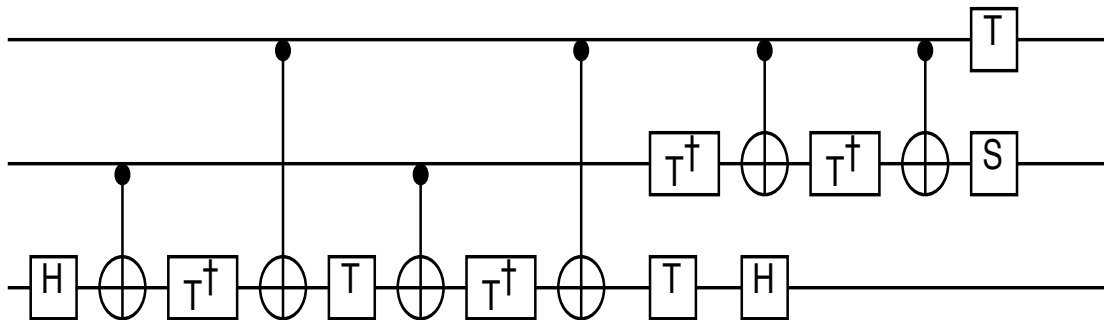


In turn, the Toffoli gate and controlled-U can be built out of CNOTs and one-bit gates.



where $\hat{C}\hat{A}\hat{B} = \hat{I}$ and $\hat{C}\hat{X}\hat{B}\hat{X}\hat{A} = \hat{U}$.

Here is a circuit for the Toffoli gate in terms of CNOTs and single-bit gates.



where S denotes the phase gate and T the $\pi/8$ gate,

$$\hat{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \hat{T} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

Note that classically it is impossible to build a Toffoli gate from two-bit and one-bit reversible gates!

It is also possible to enact the controlled^k gate using CNOTs and one-bit gates without ancillas, though it is more complicated.

We have shown how to do a two-level unitary which affects basis vectors $|1 \dots 10\rangle$ and $|1 \dots 11\rangle$, or more generally, on two basis vectors which differ in exactly one bit. We need, however, to do it on two neighboring basis states $|i\rangle$ and $|i + 1\rangle$, which will in general differ in several bits. For instance, states $|3\rangle$ and $|4\rangle$ have binary representations $|011\rangle$ and $|100\rangle$, which differ in three bits. We can get around this problem by permuting the basis vectors with a sequence of CNOTs until $|i\rangle$ and $|i + 1\rangle$ are mapped onto two basis vectors which differ in only one bit.

Let's see an example of how this works. Suppose we want to do a two-level unitary on $|3\rangle$ and $|4\rangle$, which differ in three bits: $|011\rangle$ and $|100\rangle$. We begin by doing a CNOT from one of the differing bits (say bit 0) to another (say bit 1):

$$|011\rangle \rightarrow |001\rangle, \quad |100\rangle \rightarrow |100\rangle.$$

These new basis vectors differ in two bits (bits 0 and 2). We then do this again, by doing a CNOT from bit 2 to bit 0:

$$|001\rangle \rightarrow |001\rangle, \quad |100\rangle \rightarrow |101\rangle.$$

The basis vectors now differ in only one bit. We apply an \hat{X} gate to each bit where both basis vectors are zero:

$$|001\rangle \rightarrow |011\rangle, \quad |101\rangle \rightarrow |111\rangle.$$

We can now apply a controlled²- \hat{U} gate with the controls on bits 0 and 1 and the target bit 2.

We've done the correct transform, but the basis vectors are still permuted. We now do the permuting sequence in reverse: first the \hat{X} gates, then the CNOTs; and we end up with the correct two-level unitary \hat{V} .

Of course, the other basis states have also been permuted during this process. But since the controlled² gate affects only the two basis vectors, all other effects are removed by undoing the permutation.

This is our construction: 1) We decompose the unitary \hat{U} into a product of $N(N - 1)/2$ two-level unitaries \hat{V}_i . 2) We decompose each two-level unitary into a permutation of the basis vectors, a controlled ^{$n-1$} - \hat{U} gate, and the reverse permutation. 3) We decompose the controlled ^{$n-1$} - \hat{U} gate into (e.g.) Toffoli gates and a controlled- \hat{U} gate. 4) We replace the Toffolis and controlled- \hat{U} with circuits involving only CNOTs and single bit gates. **QED**

Efficiency of circuits

This construction takes a lot of gates. For n bits, we need $O(2^{2n})$ two-level unitaries. Each of those will require $O(n)$ CNOTs and single-bit gates, giving the whole circuit a complexity of $O(n2^{2n})$.

One might imagine that this is due to the inefficiency of the construction. After all, the techniques used in an existence proof are chosen for conceptual simplicity rather than optimal efficiency. Unfortunately, this is not the case. In general, most unitary transformations on n bits really will require an exponential number of gates to perform.

However, we are fortunately not interested in the vast majority of possible unitary transformations. The algorithms we will be considering all correspond to unitaries with efficient (i.e., polynomial in n) circuits.

Approximate gates

In making this construction, we assumed that an infinite number of possible gates could be produced on demand, with whatever precision we like. This is not a very realistic assumption. In general, there may be limits to the precision with which we can perform a given unitary; and we may have only a fixed finite set of gates to draw on, rather than an infinite family.

First let us see that if our gates are not perfect, but have sufficiently good precision, then the circuits built out of them will also have sufficiently good precision.

We define the *operator norm*

$$\|\hat{O}\| = \sup_{|\xi\rangle \neq 0} \sqrt{\frac{\langle \xi | \hat{O}^\dagger \hat{O} | \xi \rangle}{\langle \xi | \xi \rangle}}.$$

For unitaries, $\|\hat{U}\| = 1$. We say two unitaries \hat{U} and \hat{U}' are *close* if $\|\hat{U} - \hat{U}'\| \leq \delta$ for some small $\delta > 0$. Because of the way the operator norm is defined, we see that this implies that their action on states is also close.

Suppose we have N operators $\hat{U}_1, \dots, \hat{U}_N$, and there are N approximations to them $\hat{U}'_1, \dots, \hat{U}'_N$ such that $\|\hat{U}_j - \hat{U}'_j\| \leq \delta_j$ for some set of small numbers $\delta_1, \dots, \delta_N$.

Then we can show that

$$\|\hat{U}_N \cdots \hat{U}_1 - \hat{U}'_N \cdots \hat{U}'_1\| \leq \sum_j \delta_j.$$

The errors accumulate (at most) linearly.

Finite sets of gates

We can make use of this result about approximate gates to show that it is possible for a finite set of gates to be universal, in the sense that it is possible to approximate any unitary transformation with arbitrary precision using only this finite set of gates.

Remember that any single bit gate can be decomposed into rotations about the X and Z axes.

$$\hat{R}_{\vec{n}}(\theta) = \hat{R}_Z(\gamma)\hat{R}_X(\beta)\hat{R}_Z(\alpha)$$

Suppose that we have gates $\hat{A} = \exp(i\eta\hat{X})$ and $\hat{B} = \exp(i\nu\hat{Z})$, where η and ν are *irrational multiples of π* .

For any angle θ there are integers m and n such that for any $\delta > 0$

$$\begin{aligned}\|\hat{A}^m - \hat{R}_X(\theta)\| &\leq \delta \\ \|\hat{B}^n - \hat{R}_Z(\theta)\| &\leq \delta.\end{aligned}$$

This works because of the following fact: if ν is an irrational multiple of π , then the angles $n \cdot \nu \bmod 2\pi$ for all integers $n > 0$ form a dense set in $[0, 2\pi)$. So for any angle θ , there is an integer n such that $n \cdot \nu \bmod 2\pi$ lies arbitrarily close to θ .

How big will m and n generally have to be to approach θ with accuracy δ ? In fact, the convergence is fairly rapid; it goes like $O(\log^c(1/\delta))$, where c is a positive constant close to 2. So, while we will still need an exponential number of gates to do a generic unitary on n bits, the additional overhead in using only a finite set of single-bit unitaries is only polynomial in n .

We see therefore that the CNOT, \hat{A} and \hat{B} gates would provide a *universal set* of gates for all quantum computations.

Standard universal sets

Rather than using gates like \hat{A} and \hat{B} , the most common universal sets use different one-bit gates. One such set, which Nielsen and Chuang call the *standard set*, includes the CNOT, Hadamard, phase and $\pi/8$ gates. One can show without too much trouble that the Hadamard and $\pi/8$ gates can generate all single-bit unitaries.

Another standard set which is sometimes used is the Toffoli, CNOT, Hadamard and phase gates. There are a number of other universal sets which have been proposed. Some are easier to do than others for particular implementations of a quantum computer.

Finally, some sets are *not* universal. In particular, the \hat{X} , \hat{Y} , \hat{Z} and \hat{H} gates are not sufficient to generate all one-bit gates.

Next time: The Quantum Fourier Transform