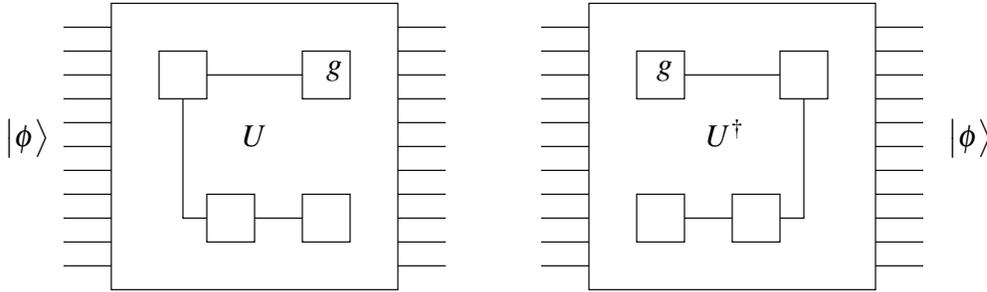


Reversible Computation

A quantum circuit acting on n qubits is described by an $2^n \times 2^n$ unitary operator U . Since U is unitary, $UU^\dagger = U^\dagger U = I$. This implies that each quantum circuit has an inverse circuit which is the mirror image of the original circuit and which carries out the inverse operator U^\dagger .



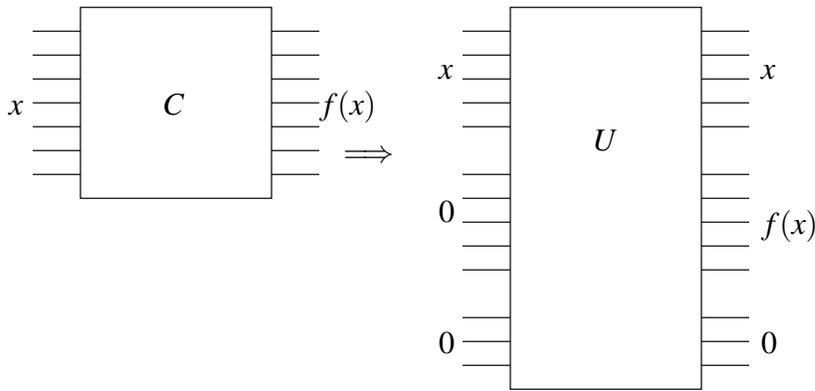
The circuits for U and U^\dagger are the same size and have mirror image gates. Examples:

$$\begin{aligned} H &= H^\dagger \\ \text{CNOT} &= \text{CNOT}^\dagger \\ R_\theta &= R_{-\theta}^\dagger \end{aligned}$$

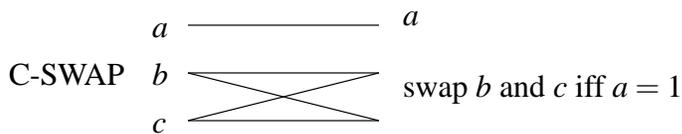
1 Simulating Classical Circuits

Let us first consider whether given any classical circuit there is an equivalent quantum circuit. More concretely, suppose there is a classical circuit that computes a function $f(x) \in \{0,1\}^m$ on input $x \in \{0,1\}^n$, is there a quantum circuit that does the same? Obviously such a quantum circuit must map computational basis states to computational basis states (i.e. it must map each state of the form $|x\rangle$ to the state $|f(x)\rangle$). A unitary transformation taking basis states to basis states must be a permutation. (Indeed, if $U|x\rangle = |u\rangle$ and $U|y\rangle = |u\rangle$, then $|x\rangle = U^{-1}|u\rangle = |y\rangle$.) Therefore $m = n$ and the function $f(x)$ must be a permutation on the n -bit strings. Since this must hold after every application of a quantum gate, it follows that if a quantum circuit computes a classical function, then it must necessarily be reversible.

How can a classical circuit C which takes an n bit input x and computes $f(x)$ be made into a reversible quantum circuit that computes the same function? The circuit must never lose any information, so how could it compute a function mapping n bits to $m < n$ bits (e.g. a boolean function, where $m = 1$)? The way to achieve this is to require that the circuit output both the input x and the output $f(x)$. In addition, the quantum circuit may need some additional scratch qubits during the computation since individual gates can't lose any information either. The consequence of these constraints is illustrated below.



How is this done? Recall that any classical AND and OR gates can be simulated with a C-SWAP gate and some scratch $|0\rangle$ qubits.



If we construct the corresponding reversible circuit RC, we have a small problem. The CSWAP gates end up converting input scratch bits to garbage. How do we restore the scratch bits to 0 on output? We use the fact that RC is a reversible circuit. The sequence of steps for the overall circuit is

$$(x, 0^k, 0^m, 0^k, 1) \xrightarrow{C'} (x, y, \text{garbage}_x, 0^k, 1) \xrightarrow{\text{copy}_y} (x, y, \text{garbage}_x, y, 1) \xrightarrow{(C')^{-1}} (x, 0^k, 0^m, y, 1) .$$

Overall, this gives us a clean reversible circuit \hat{C} corresponding to C .

