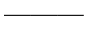
Mathematics of Quantum Computation II

Wim van Dam
Department of Computer Science, University of California at Santa Barbara, Santa Barbara, CA 93106-5110, USA

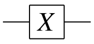
Notes for the graduate course “Quantum Computation and Quantum Information” (290A), Spring 2005. v1

Standard Quantum Gates: The following gates are standard single qubit gates:

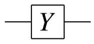
$$\text{Identity:}\quad I=\begin{pmatrix}1&0\\0&1\end{pmatrix}$$



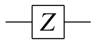
$$\text{N{\scriptsize OT}:}\quad X=\begin{pmatrix}0&1\\1&0\end{pmatrix}$$



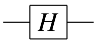
$$Y=\begin{pmatrix}0&-\mathrm{i}\\\mathrm{i}&0\end{pmatrix}$$



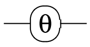
$$Z=\begin{pmatrix}1&0\\0&-1\end{pmatrix}$$



$$\text{Hadamard:}\quad H=\frac{1}{\sqrt{2}}\begin{pmatrix}1&1\\1&-1\end{pmatrix}$$

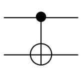


$$\theta\text{-Phase Rotation:}\quad R_z(\theta)=\begin{pmatrix}1&0\\0&\mathrm{e}^{\mathrm{i}\theta}\end{pmatrix}$$

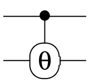


Typically, two qubit gates are of the kind where a *control bit* determines whether or not a single qubit operation is applied to the *target bit* or not.

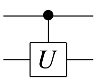
$$\text{CN{\scriptsize OT}}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$



$$C\text{-}R_z(\theta)=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&\mathrm{e}^{\mathrm{i}\theta}\end{pmatrix}$$

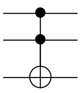


$$\text{Controlled-}U=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&U_{11}&U_{12}\\0&0&U_{21}&U_{22}\end{pmatrix}$$



The three qubit, CCNOT gate is crucial for the implementation of classical, reversible computation.

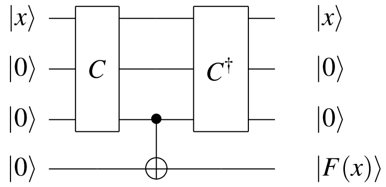
$$\text{CCN{\scriptsize OT}}=\begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{pmatrix}$$



Circuit Calculus: Each quantum gate has a matrix representation, and we assume that the ordering of the dimensions is determined by the alphabetical ordering on the bit strings $\{0,1\}^n$. (See the Exercises of the course.) The joint behavior of gates that work in parallel is calculated with the help of tensor products, while sequential gates are expressed by matrix products. In both cases one should pay attention to the order of multiplication.

Universal Reversible Computation: With a CCNOT gate we can implement an AND-operation by CCNOT : $|a,b,0\rangle\mapsto|a,b,ab\rangle$ for all $a,b\in\{0,1\}$. Hence, using CCNOT and NOT gates, we can implement any Boolean function $F:\{0,1\}^n\to\{0,1\}$ as efficient as is possible classically. However, such an implementation will not erase the input value x and typically also produces ‘garbage’ bits that are a side-effect of the computation. (Take for example the computation of $F(x,y,z)=xyz$ by the sequence of transformations $|x,y,z,0,0\rangle\mapsto|x,y,z,xy,0\rangle\mapsto|x,y,z,xy,xyz\rangle=|x,y,z,xy,F(x,y,z)\rangle$, which has as garbage the intermediate bit value xy.) Because of the reversibility requirement it is impossible to get rid of the input bits, but it is possible to erase the garbage bits as follows.

Note that each quantum circuit can be reversed. Hence, if there is a circuit C that implements the transformation $|x,0,0\rangle\mapsto|x,g_x,F(x)\rangle$ (with g_x the garbage bits specifically for the input $x\in\{0,1\}^n$), then the inverse circuit $C^{-1}=C^\dagger$ implements the mapping $|x,g_x,F(x)\rangle\mapsto|x,0,0\rangle$. Now, by applying a CNOT between C and $C^\dagger$ we get the following circuit



which implements the desired transformation $|x,0,0,0\rangle\mapsto|x,0,0,F(x)\rangle$ for all x.

Note that this construction applies to all possible quantum circuits C, which gives us the following important result.

Clean Quantum Computation Theorem: If there is a quantum circuit C that implements the unitary mapping $|x,0,0\rangle\mapsto|x,g_x,F(x)\rangle$ for a Boolean function $F:\{0,1\}^n\to\{0,1\}^m$, then there exists a quantum circuit (only twice the size of C) that implements the clean computation $|x,0,0\rangle\mapsto|x,0,F(x)\rangle$. Notice that it is crucial for this construction that we have clean working qubits bits lying around that we can use during the computation.

Acknowledgment: The circuits in these exercises were drawn using the Q-circuit LaTeXpackage of Bryan Eastin and Steven T. Flammia.