Lecture & Proseminar 250078/250042
"Quantum Information, Quantum Computation, and Quantum Algorithms" WS 2023/24

— Exercise Sheet #7 —

Problem 18: Circuits for one-qubit unitaries and controlled unitaries.

Let $R_\alpha(\phi) = e^{i\phi/2\,\sigma_\alpha}$, $\alpha = x, y, z$.

1. Show that for any H with $H^2 = I$, $e^{i\vartheta H} = \cos(\vartheta)\, I + i\, \sin(\vartheta)\, H$. (Recall that exponentials of operators are defined through the Taylor series.)

2. Show that any one-qubit unitary U can be written as

$$U = e^{i\phi} R_z(\alpha) R_x(\beta) R_z(\gamma) \ .$$

   Construct the angles α, β, γ, and ϕ explicitly in terms of U. (It can be helpful to start by choosing a suitable parametrization of the entries of U.)

3. Show that also such a decomposition of the form

$$U = e^{i\phi'} R_z(\alpha') R_y(\beta') R_z(\gamma') \tag{1}$$

   exists.

4. Use (1) to show that for a special unitary 2×2 matrix $U \in \mathrm{SU}(2)$ (i.e. $\det(U) = 1$), there exist matrices $A, B, C \in \mathrm{SU}(2)$ such that $ABC = I$ and $AXBXC = U$, where X is the Pauli x matrix. (*Hint:* Try to split up the individual rotations in (1) into several rotations, e.g. $R_z(\alpha') = R_z(\alpha' + \delta) R_z(-\delta)$, and use the fact that commutation with X changes the rotation direction of y and z rotations, e.g. $X R_z(\delta) = R_z(-\delta) X$.)

5. Use this to construct a circuit which implements a controlled-U gate (for *any* unitary U), which uses the matrices A, B, and C, CNOT gates, and an additional one-qubit gate E which adjusts relative phases.

Problem 19: Reversible classical 2-bit gates.

1. Show that all reversible classical 2-bit gates $G(x_1, x_2) = (y_1, y_2)$ can be written as a linear map over $\mathbb{Z}_2$, i.e.,

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = M \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mod 2$$

   (where $x_i, y_i \in \{0, 1\}$, and M has entries 0 and 1).

2. Show that all those gates can be decomposed into only NOT and CNOT gates. (A useful identity can be that three consecutive CNOTs with opposite alignment swap the input bits.) Of course, you can solve this question before question 1, and then just show that CNOT and NOT are linear maps over $\mathbb{Z}_2$.

3. Show that this implies that any classical circuit consisting only of reversible 2-bit gates can be written as a linear transformation over $\mathbb{Z}_2$.

4. Show that the Toffoli gate is not of this form – that is, reversible classical two-bit gates are not universal for classical computation.

(*Note:* The class of problems which can be solved this way in time poly(n), with n the number of bits, defines the complexity class $\oplus$L (pronounced "Parity-L"). $\oplus$L can be simulated in time $\log(n)^2$ by a general classical circuit, and is thus indeed much more restricted than general efficient classical computations, which can have a runtime poly(n).)

Problem 20: n-qubit Toffoli gates.

An n-qubit Toffoli gate is a Toffoli gate with $n-1$ controls; i.e., it flips the n'th bit if and only if the other $n-1$ bits are all one. The goal of this problem is to see how n-qubit Toffolis can be built up from simpler gates, most importantly normal 3-qubit Toffolis.
The subsequent constructions rely on using ancilla qubits. For all problems below, **consider two cases**:

- First, the ancillas are initialized in the state $|0\rangle$.
- Second, the ancillas are in some unknown state $|\phi\rangle$.

In both cases, we want to return the qubit in the state in which it was initially. While the first case is of course covered by the second case, you should also consider whether there is a simpler realization in the first case.
(Being able to realize the gate using an unknown ancilla which is returned in the same state is very useful, since then any qubit on which the gate to be constructed does not act can serve as a "temporary" ancilla.)

1. Show that the n-qubit Toffoli gate can be implemented using two $n-1$-qubit Toffoli gates and two regular 3-qubit Toffoli gates using one ancillary qubit.
2. Using the previous procedure to recursively decompose every gate into 3-qubit Toffoli gates, how many 3-qubit Toffoli gates do you need to construct the n-qubit Toffoli gate? How many ancillas are needed? (Are there ways to save ancillas?)
3. Find a construction which is more efficient in terms of the scaling of the number 3-qubit Toffoli gates used, at the cost of using more ancillas. (You should get a circuit which requires a number of 3-fold Toffoli gates which scales linearly with n.)

(*Hint:* Remember that the Toffoli gate can be used to build a logical AND gate using ancillas.)