Problem set 3 for 41076: Methods in Quantum Computing

due at 11:59 pm of October 9th 2022
15 points

1 Paulis (6 points)

Consider the Pauli matrices $\sigma_1 = X$, $\sigma_2 = Y$ and $\sigma_3 = Z$

1. Show that the anticommutator $\{A, B\} = AB + BA$ for Paulis satisfies

$$\{\sigma_j, \sigma_k\} = 2\delta_{j,k}\mathcal{I} \tag{1}$$

   where $\delta_{i,j}$ is the Kronecker delta, $\delta_{i,j} = 1$ if $j = k$ and 0 otherwise.

2. Show that the commutator $[A, B] = AB - BA$ for Paulis satisfies

$$[\sigma_j, \sigma_k] = i2\epsilon_{j,k,l}\sigma_l \tag{2}$$

   Here we use Einstein notation ($\epsilon_{j,k,l}\sigma_l$ stands for $\sum_l \epsilon_{j,k,l}\sigma_l$) and the Levi-Civita ϵ symbol in 3 dimensions. For Levi-Civita, $\epsilon_{1,2,3} = 1$ and exchanging any two indices satisfies $\epsilon_{j,k,l} = -\epsilon_{k,j,l}$ (completely antisymmetric). Thus, $\epsilon_{2,1,3} = \epsilon_{1,3,2} = \epsilon_{3,2,1} = -1$. because of the antisymmetry, $\epsilon_{j,k,l}$ will be zero if any of the indices are equal to each other, i.e. $\epsilon_{1,1,2} = 0$.

3. Show that

$$\sigma_j\sigma_k = \delta_{j,k}\mathcal{I} + i\epsilon_{j,k,l}\sigma_l. \tag{3}$$

   This is a useful identity whenever one needs to multiply a lot of Paulis.

4. Consider Paulis on one qubit

$$P = \{\pm\mathcal{I}, \pm X, \pm Y, \pm Z, \pm i\mathcal{I}, \pm iX, \pm iY, \pm iZ\} \tag{4}$$

   Show that P with multiplication forms a group.

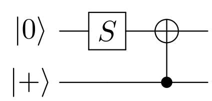
2 Clifford gates (5 points)

This problem will demonstrate how stabilizers can be used used for efficiently performing certain quantum computation.

1. Consider a state $|\psi\rangle = |0\rangle|+\rangle$. Find two two-qubit Paulis P_1, P_2 (other than identity or its multiples) that stabilize this state, i.e. $P_1|\psi\rangle = P_2|\psi\rangle = |\psi\rangle$.

2. Compute $SZS^\dagger$ where S is the phase gate $S = \begin{pmatrix} 1 & 0 \\ 0 & e^{-\frac{i\pi}{2}} \end{pmatrix}$

3. Use stabilizers (i.e. not quantum states) to simulate the following circuit. What will be the stabilizers corresponding to the output?



3 Complexity of a quantum algorithm (4 points)

The goal of this problem is to estimate the gate complexity and query complexity of a quantum algorithm for computing gradient. Let $f : \mathcal{R}^d \to \mathcal{R}$ be a function of d variables. We would like to estimate its gradient $\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_d}\right)$ up to an error $\frac{1}{\epsilon^2}$.

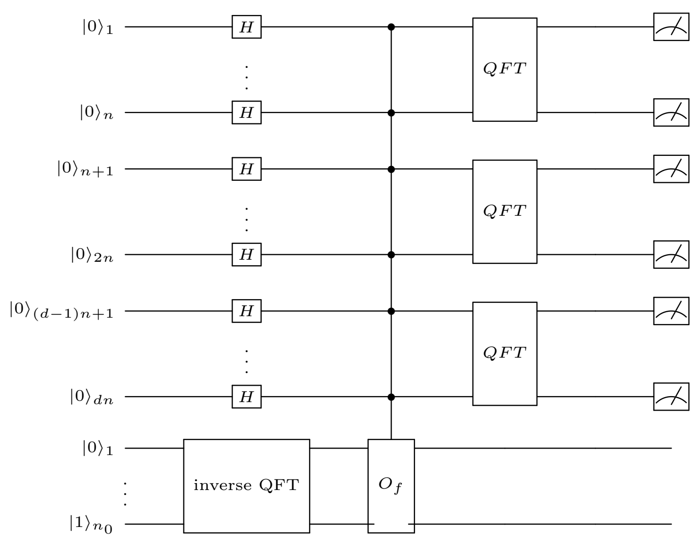


Figure 1: A quantum gradient algorithm by Jordan.

1. First, we estimate the complexity of a classical algorithm. For each component of the gradient $\frac{\partial f}{\partial x_i}$, we evaluate the function f in points $f(x_0, \ldots, x_i + \epsilon/2, \ldots, x_d)$ and $f(x_0, \ldots, x_i - \epsilon/2, \ldots, x_d)$ and compute $\frac{\partial f}{\partial x_i} \approx \frac{f(x_0,\ldots,x_i+\epsilon/2,\ldots,x_d) - f(x_0,\ldots,x_i-\epsilon/2,\ldots,x_d)}{\epsilon}$. How many queries to f does this algorithm use? A tight bound in big-O notation is sufficient.

2. A quantum algorithm accesses f through an oracle O_f such that $O_f|x_1, \ldots, x_d\rangle|z\rangle = |x_1, \ldots, x_d\rangle|z \oplus f(x)\rangle$ where each register x_i consist of n qubits and z a real number encoded into n_0 qubits. The circuit for approximating the gradient is depicted in Fig. 1. It uses d input register with n qubits each and an output register with n_0 qubits. First, it performs a Hadamard transform on the input registers and inverse quantum Fourier transform on the output register. Next, we apply the oracle and lastly apply QFT on each input register. Compute the gate and query complexity of the algorithm. Big-O asymptotic result in terms of d, n and n_0 is sufficient.

While this algorithm is indeed very interesting, it does not have a lot of applications in quantum computing because the oracle O_f is very powerful and its implementation often wipes out the speedup.