Quantum Computation and Error Correction: Exercise Sheet 2

Submit before 04/11, 4 p.m.

Problem 1. Universal quantum computing: We want to show that the gate set $\{CNOT, H, T\}$ is universal, meaning we can approximate any arbitrary unitary gate to any desired accuracy using only these three gates in an n-qubit quantum circuit. Here, we will only focus on the following problem statement: *'How does one achieve an arbitrary single-qubit unitary operation?'* The approximation of general n-qubit gates then follows from the known fact that the $CNOT$ gate, along with arbitrary single-qubit gates, is universal.

- **Problem 1.1.** (2 marks) Consider a $\frac{\pi}{4}$ rotation around the $\hat{z}$-axis (T) and a $\frac{\pi}{4}$ rotation around the $\hat{x}$-axis (HTH). Combine these operations (i.e., evaluate $THTH$) to **show** that the result is a rotation $R_{\hat{n}}(\theta)$, where $\vec{n} = \{\cos(\pi/8), \sin(\pi/8), \cos(\pi/8)\}$ and $\theta = \cos^{-1}(\cos^2(\pi/8))$.

- **Problem 1.2.** (1 mark) **Show** that repeatedly applying $R_{\hat{n}}(\theta)$ can approximate any amount of rotation about the axis $\hat{n}$. *Hint: Show that (i)* $R_{\hat{n}}(\theta)^k = R_{\hat{n}}(\theta_k)$*, providing an expression for* θ_k*, and (ii) that if* $\theta_k = \theta_{k'} \pmod{2\pi}$*, then* $k = k'$*.*

- **Problem 1.3.** (2 marks) It can be shown that any single-qubit unitary operation U can be decomposed as:

  $$U = R_{\hat{n}}(\theta_1) R_{\hat{m}}(\theta_2) R_{\hat{n}}(\theta_3)$$

  (this is analogous to Euler rotations). The second axis of rotation, $\hat{m}$, can be obtained by applying a Hadamard gate to the first axis: $R_{\hat{m}}(\theta) = H R_{\hat{n}}(\theta) H$. **Show** that an arbitrary unitary operation on a single qubit can be expressed as:

  $$U = R_{\hat{n}}(\theta)^{n_1} H R_{\hat{n}}(\theta)^{n_2} H R_{\hat{n}}(\theta)^{n_3}$$

  where n_1, n_2, n_3 are integers.

- **Problem 1.4.** (5 marks) **Implement** in Python a $\pi/10$ rotation along the Z-axis with a distance of less than 0.01 radians between the target and the approximated rotation. To compute this distance, you may use the following function:

  ```
  import numpy as np
  from numpy import arccos as acos

  def distance(U, V):
      F = abs(np.trace(U.conj().T @ V)) / 2.0
      F = min(1.0, max(0.0, F))
      return acos(F)
  ```

  where U and V are the target and approximated rotations, respectively.

- **Problem 1.5.** (1 mark bonus) **Discuss** the practicality of this scheme as the target precision increases.

Problem 2. Querying algorithm for a 2-to-1 function: Let f be a 2-to-1 function that maps a length-n binary string to a length-m binary string, such that two different inputs x and y have the same image if and only if there is some binary string c where $y = x \oplus c$. Note that if c is the zero bitstring, then f is 1-to-1. The problem is to find the most efficient algorithm to determine c (which may be the zero string), given an oracle for f.

- **Problem 2.1.** (0.5 marks) Consider the following example with length-3 binary strings for the input:

x	000	001	010	011	100	101	110	111
$f(x)$	1010	0100	0110	1000	0110	1000	1010	0100

  Give the value of c for this example. *Note that the output bitstrings $f(x)$ do not need to be of the same length as the input bitstrings x, as the example suggests.*

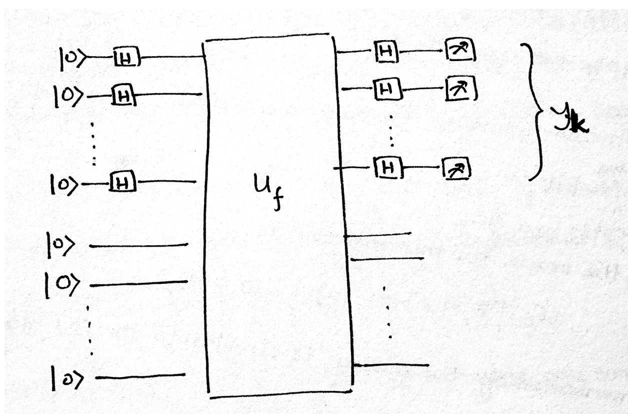
- **Problem 2.2.** (0.5 marks) **Estimate** the complexity of a classical solution for a length-n binary string.

- **Problem 2.3.** (1+1+0.5 marks) The quantum (boolean) oracle for the function f is defined as:

$$U_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle,$$

  where the first and second registers may not have the same number of qubits.

  A query consists of the following algorithm:



  Step 1: Start with two registers of n and m qubits, respectively, all initialized to the $|0\rangle$ state: $|\psi_1\rangle = |0^{\otimes n}\rangle|0^{\otimes m}\rangle$.

  Step 2: Apply the Hadamard gate to each qubit in the first register: $|\psi_2\rangle = (H^{\otimes n} \otimes I^{\otimes m})|\psi_1\rangle$ (I is the identity operator).

  Step 3: Apply the oracle: $|\psi_3\rangle = U_f|\psi_2\rangle$.

  Step 4: Apply the Hadamard gate to each qubit in the first register again: $|\psi_4\rangle = (H^{\otimes n} \otimes I^{\otimes m})|\psi_3\rangle$.

  Calculate $|\psi_4\rangle$ and the probability of measuring the state $|k\rangle$ in the first register for a generic function f.
  Then, **simplify** the expression using the fact that for a given $f(j)$, at most two terms, j and $j \oplus c$, contribute.
  Show that any bitstring y_k obtained by measuring the first register satisfies $y_k \cdot c = 0 \pmod 2$.

- **Problem 2.4.** (1 mark) *Classical post-processing.*

  We say that y_k is independent of $\{y_1, y_2, \ldots, y_{k-1}\}$ if there is no set $\{\epsilon_i \in \{0,1\}\}$ such that $y_k = \bigoplus_{i=1}^{k-1} \epsilon_i y_i$. For bitstrings of length n, it follows that there can be at most n independent bitstrings.

If we perform k queries, there is a probability p_k of finding n independent bitstrings from the results $\{y_k\}$, with $p_k > 0$ if and only if $k > n - 1$.
Assuming we have n independent bitstrings in $\{y_k\}$, **find** an efficient classical algorithm to deduce c. What is its complexity?

- **Problem 2.5.** (0.5 marks) **Estimate** the total time complexity for this hybrid quantum-classical algorithm (the quantum part + the classical post-processing) to solve the problem with a probability p. Compare this with your answer for the purely classical algorithm.
- **Problem 2.6.** (5 marks) **Implement** the quantum algorithm in Qiskit.
- **Problem 2.7.** (1 mark bonus) **Conclude** on the effectiveness of this quantum algorithm compared to the classical approach.