Quantum Computation and Error Correction: Exercise Sheet 2

Hand over before the 04/11, 4pm.

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

- **Problem 1.1.** (2 marks) Consider $\frac{\pi}{4}$ rotation around $\hat{z}$ (T) and $\frac{\pi}{4}$ rotation around $\hat{x}$ (HTH). Combine (i.e. look at $THTH$) these operations 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 repeating $R_{\hat{n}}(\theta)$ approximates any amount of rotation about the axis $\hat{n}$. *Hint: show that (i) $R_{\hat{n}}(\theta)^k = R_{\hat{n}}(\theta_k)$ where you would give θ_k, and (ii) that $\theta_k = \theta_{k'}$ mod 2 implies $k = k'$.*

- **Problem 1.3.** (2 marks) It can be shown that any unitary operation U for one qubit 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's rotation). The second axis of arbitrary rotation $\hat{m}$ can be easily deduced by applying Hadamard to the first one: $R_{\hat{m}}(\theta) = HR_{\hat{n}}(\theta)H$. **Show** that an arbitrary unitary operation on a single qubit is then given by,

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

  where n_1, n_2, n_3 are integers.

- **Problem 1.4.** (5 marks) **Implement** in python for a $\pi/10$ rotation along the Z axis within a distance of 0.01 radian between the target and approximated rotation. To compute this distance, you may use:

  ```
  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) **Conclude** on the practicality of the 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 length-n binary string, such that two different arguments x and y have the same image if and only if there is some binary string c such that $y = x \oplus c$. Note that if c is the bitstrings made of only zeros, then f is actually 1-to-1. The problem we are trying to solve is that if we have an oracle for f, then what is the best algorithm we can imagine to find c (which may be the zero string)?

- **Problem 2.1.** (0.5 marks) Lets's see an example with a length-3 binary string:

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

  Give the value of c in this example. *Note that the image bitstrings $f(x)$ do not need to be of the same size as the arguments bitstrings x, as the example suggests.*

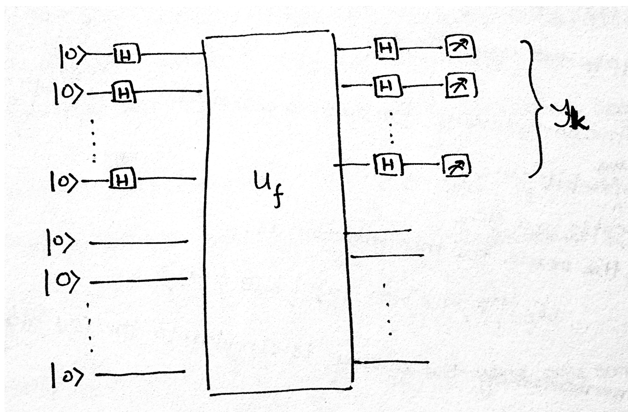
- **Problem 2.2.** (0.5 marks) **Estimate** the complexity for such a classical solution for the case of length-n binary string.

- **Problem 2.3.** (1+1+0.5 marks) The quantum (boolean) oracle for the function f verifies,

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

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

  We call a query the following algorithm:



  step 1: start with two registers of n-qubits and m-qubits respectively, all initialized in the $|0\rangle$ state: $|\psi_1\rangle = |0^{\otimes n}\rangle|0^{\otimes n}\rangle$,

  step 2: apply the many-Hadamard gate to first register: $|\psi_2\rangle = H^{\otimes n} \otimes I^{\otimes m}|\psi_1\rangle$ (I being the identity),

  step 3: apply the oracle: $|\psi_3\rangle = U_f|\psi_2\rangle$

  step 4: apply the many-Hadamard gate to 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 f.
  Then **simplify** the expression with the fact that only up to two terms, j and $j \oplus c$, would contribute to a given $f(j)$.
  Show that any bitstrings y_k obtained by measuring the first register satisfy $y_k \cdot c = 0 \bmod 2$.

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

  We say that y_k is independent from $\{y_1, y_2, ..., y_{k-1}\}$ if there is no $\{\epsilon_k = 0, 1\}$ such that $y_k = \oplus_{i=1}^{k-1} y_i$. For bitstrings of length n, it follows that there is at most n independent bitstrings. If we perform k queries, there is a probability p_k of finding n independents bitstrings from the results $\{y_k\}$, with $p_k > 0$ if and only if $k > n - 1$.
  Assuming that there are n independents bitstrings in $\{y_k\}$, **find** a classical algorithm that efficiently deduce c. What is its complexity?

- **Problem 2.5.** (0.5 mark) **Estimate** the total time complexity for the hybrid quantum algorithm (the quantum part + the classical post-processing), to solve the problem with a probability p. Compare with your answer for the purely classical algorithm.

- **Problem 2.6.** (5 marks) **Implement** the quantum algorithm in Qiskit.

- **Problem 2.6.** (1 bonus mark) **Conclude.**