Comp Sci 880 HW1

Hongtao Zhang

March 18, 2023

Contents

1. Question 1 2
2. Question 2 3
   2.1. Balanced Function 3
   2.2. Constant function 4
3. Question 3 5
4. Question 4 6
5. Question 5 7
   5.1. Domain 7
   5.2. Range 7

1 Question 1

What we want to do is to split the real part and the imaginary part, and then do calculation with only real entry but resulting complex effect.

Firstly we need to split the amplitude into two part. It is easy to check that the RHS also have 2-norm 1.

$$\begin{pmatrix} a+ci \\ b+di \\ 0 \\ 0 \end{pmatrix} \to \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \tag{1}$$

(Based on Piazza post, we can assume this transformation is given)

Then we can do the calculation with only real entry, and then combine the result back.

For any Complex Unitary $U_c = A + Bi$, we can represent it with a real unitary matrix U as

$$U = \begin{pmatrix} A & -B \\ B & A \end{pmatrix}$$

Then its action on the state vector $\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$ is exactly the same as the action of U on the state vector $\begin{pmatrix} a+ci \\ b+di \\ 0 \\ 0 \end{pmatrix}$ with the transformation (1).

The thing left to check is that the matrix U is unitary.

Proof.

$$\begin{aligned} UU^* &= \begin{pmatrix} A & -B \\ B & A \end{pmatrix} \begin{pmatrix} A^* & B^* \\ -B^* & A^* \end{pmatrix} \\ &= \begin{pmatrix} AA^* + BB^* & 0 \\ 0 & AA^* + BB^* \end{pmatrix} \end{aligned}$$

We know that matrix $U_c = A + Bi$ is unitary, so $U_c U_c^* = I = (A + Bi)(A + Bi)^T$

$$\begin{aligned} (A+Bi)(A+Bi)^* &= (A+Bi)(A^* - B^*i) \\ &= AA^* - AB^*i + A^*Bi - B^*iBi \\ &= AA^* + BB^* \\ &= I \end{aligned}$$

Therefore, $AA^* + BB^* = I$, which means U is unitary. □

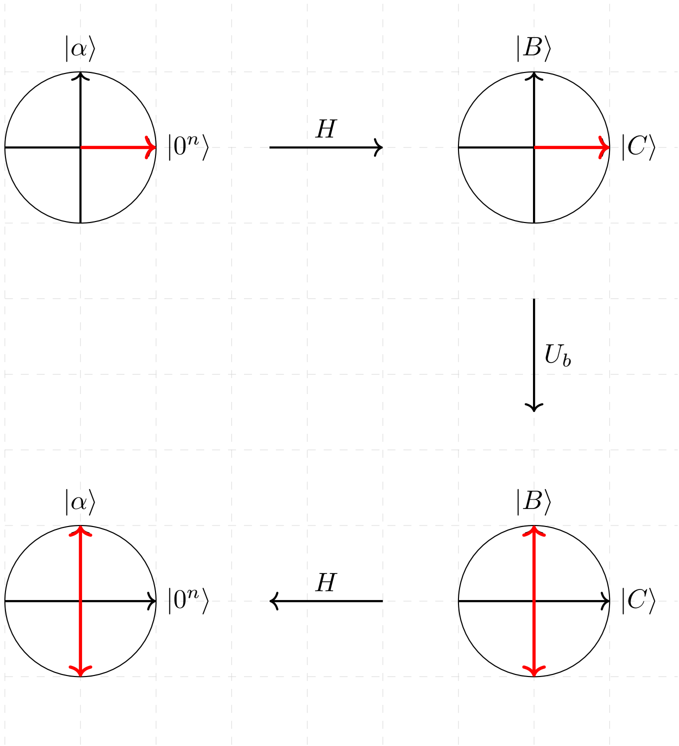
Finally, if we do a measurement on the first m qubits when the ancillary qubits are in state $|0\rangle$, we will get the same result for the complex gates on the first m qubits. (**Need check**)

2 Question 2

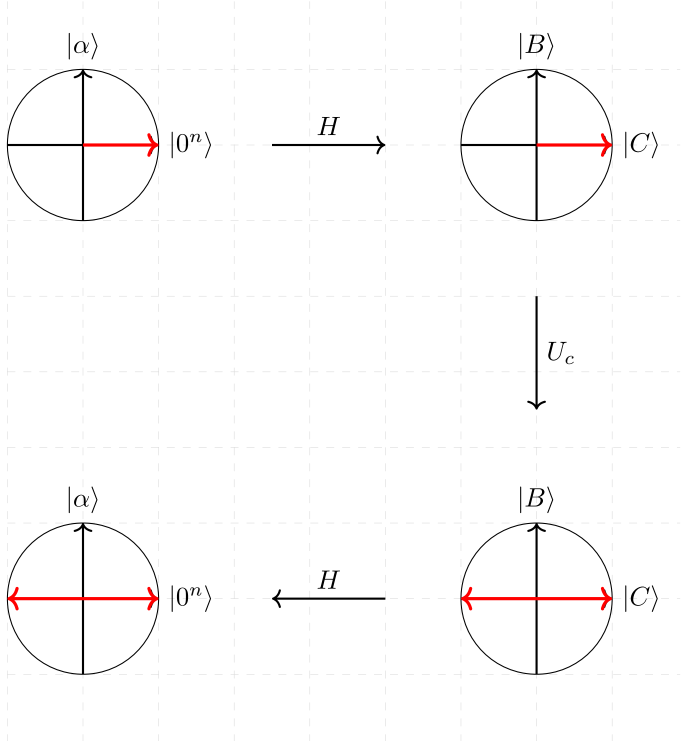
For Deutsch-Jozsa, we firstly apply Hadamard gate on all qubits, then apply the oracle, and then apply Hadamard gate on the first qubit.

Then we can view the Hadamard gate as the transformation to the amplitude transformation, and the oracle as the transformation to the phase transformation, and the other Hadamard gate as the transformation back to original basis.

2.1 Balanced Function



2.2 Constant function



3 Question 3

We can create a controlled version of the black box so make sure the black box transform to a state that is orthogonal to each other.

In the solution in error elimination, we have learned how to create a controlled version of the phase kick back black box. We can use this to create a controlled version of the black box.

The idea is to have a black box that is controlled by an additional ancilla that has uniform probability between $|0\rangle$ and $|1\rangle$.

Then we can just apply the same analysis as in the Deutsch-Jozsa algorithm.

In this case, the black box will be applied with probability $\frac{1}{3}$.

If the black box is the constant case (i.e. $2^n 0$), then the resulting distribution of the modified black box is identical to the original black box, with up to a global phase change.

If the black box is with 2^{n-2} 0 and 2^{n-1} 1, then the resulting distribution of the modified black box will be

$$P(x=0) = p + \frac{1}{4} \cdot (1-p) = \frac{1}{3} + \frac{1}{4} \cdot (1 - \frac{1}{3}) = \frac{1}{3} + \frac{1}{6} = \frac{1}{2}$$

Therefore, the resulting distribution is uniform, so the black box is balanced.

Therefore, we can follow the same algorithm as in the Deutsch-Jozsa algorithm.

4 Question 4

TODO!

5 Question 5

There are two idea about how to solve this problem in $n\sqrt{N}$.

One from the domain and one from the range.

Note that $n = \log N$, so this follows divide and conquer paradigm.

5.1 Domain

We can use the domain idea to solve this problem.

5.2 Range

The idea is very simple and like a binary search, each time we chunk half of the range, and see whether we can find the element in the chunk.

If we found, then the lowest index is in the chunk, otherwise it is in the other chunk.

We will need to do this n time.

Each time Grover search will take $O(\sqrt{N})$ time, so the final query complexity is $O(n\sqrt{N})$.