Quantum Algorithms via Linear Algebra

— Deutsch's Algorithm

Dao-Yun Xu

College of Computer Science and Technology, Guizhou University

Outline

7. Phil's Algorithm

8. Deutsch's Algorithm

9. The Deutsch-Jozsa Algorithm

7. Phil's Algorithm

Each algorithm will be presented as computing a series of vectors. The goal of Phil's Algorithm is to give the schema presenting quantum algorithms. Formally, it is of form:

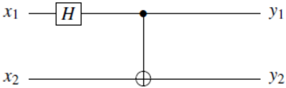
"Given an X, the algorithm finds a Y within time Z"

by a series of explicitly vectors from a start vector to last vector.

Then we will understand what the result of the last step of the algorithm does because in all cases the last step is a quantum measurement.

To understand measurements, we must know the amplitudes given by the last vector because the measurement returns k with the amplitude squared of the k-th coordinate.

A Two-Qubit Example



Phil carries out the composition of $V_1 = H \otimes I$ and $V_2 = CNOT$.

We index vectors in this two-qubit space by xy, where x and y are single bits.

Algorithm:

(1). The initial vector is a_0 so that $a_0(00) = 1$,i.e., $a_0 = e_{00}$.

(2). The next vector a_1 is the result of applying the Hadamard transform on qubit line 1 only.

(3). The final vector a_2 is the result of applying $CNOT$ to a_1.

Analysis

(1) $a_0 = e_{00}$.

(2) $a_1 = \frac{1}{\sqrt{2}}(e_0 + e_1) \otimes e_0$

$$= \frac{1}{\sqrt{2}}(e_{00} + e_{10})$$

$$= \frac{1}{\sqrt{2}}[1, 0, 1, 0]^T.$$

(3) By $CNOT\ e_{00} = e_{00}$, $CNOT\ e_{10} = e_{11}$, we have

$$a_2 = \frac{1}{\sqrt{2}}(e_{00} + e_{11}) = \frac{1}{\sqrt{2}}[1, 0, 0, 1]^T.$$

So far, we have not said anything about taking measurements — instead, we are able to specify the final pure quantum state:

$$\frac{1}{\sqrt{2}}(e_{00} + e_{11}).$$

Entangled pairs:

In the Hilbert space coordinates it doesn't look exciting, but let's interpret it back in the quantum coordinates:

$$a_2 = \frac{1}{\sqrt{2}}(e_{00} + e_{11}).$$

This state is pure and not a tensor product of two other states, so it is entangled.

If we measure both qubits, then we will only get 00 or 11, never 01 or 10.

If we measure just the first quantum coordinate and get 0, then we know already that any measurement of the second quantum coordinate will give 0.

Thus, Phil's algorithm has produced an entangled pair of qubits. This is basics of qutantum communication.

Bell's states: (EPR entangle pairs)

$a_2 = (CNOT \cdot H)a_0$:

a_0	a_2		$Beel's\ states$
e_{00}	$\frac{1}{\sqrt{2}}(e_{00} + e_{11})$	$\frac{1}{\sqrt{2}}[1,0,0,1]^T$	β_{00}
e_{01}	$\frac{1}{\sqrt{2}}(e_{01} + e_{10})$	$\frac{1}{\sqrt{2}}[0,1,1,0]^T$	β_{01}
e_{10}	$\frac{1}{\sqrt{2}}(e_{00} - e_{11})$	$\frac{1}{\sqrt{2}}[1,0,0,-1]^T$	β_{10}
e_{11}	$\frac{1}{\sqrt{2}}(e_{01} + e_{10})$	$\frac{1}{\sqrt{2}}[0,1,-1,0]^T$	β_{11}

EPR entangled pair β_{00} and qutantum communication:

The EPR entangle pair $\beta_{0,0}$ becomes significant:

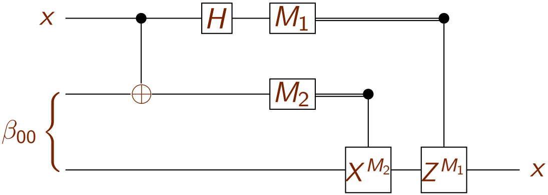
When we are able to give the first qubit to "Alice"' sitting 10 miles east of Lake Geneva and the second qubit to her friend "Bob" sitting 10 miles west, and each does a measurement at instants such that no signal of Alice's result can reach Bob before he measures and vice versa.

Whatever result Alice gets, Bob gets too. It seems that is not involved in distance between Alice and Bob.

However, we do not have to think in physical terms — Phil's output is just an ordinary vector in our four-dimensional Hilbert space. What we do need to finish the analysis of our algorithms, is measurement.

Application of β_{00}: Quantum cipher

For a single qubit $|x\rangle = (\alpha|0\rangle + \beta|1\rangle) = (\alpha e_0 + \beta e_1)$:

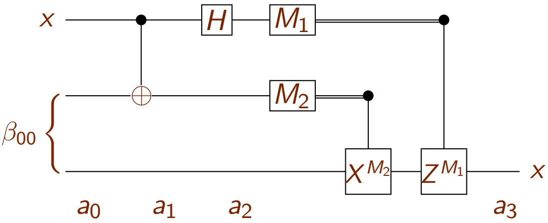


$$a_0 = (\alpha e_0 + \beta e_1) \otimes \beta_{00} = (\alpha e_0 + \beta e_1) \otimes \frac{1}{\sqrt{2}}(e_{00} + e_{11});$$

where $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

The first two qubits is for Alice, and the third qubit is for Bob.

Application of β_{00}: Quantum cipher



Alice measures first two qubits, and Bob read the result of measuring third qubit by Alice's measurements, which is one of the following four results:

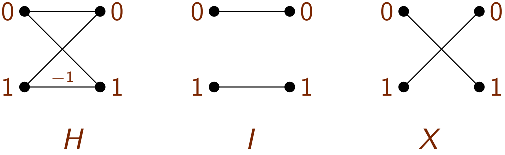
$$\begin{array}{lcl} 00 & \mapsto & a_3(00) \equiv (\alpha e_0 + \beta e_1) \\ 01 & \mapsto & a_3(01) \equiv (\alpha e_1 + \beta e_0) \\ 10 & \mapsto & a_3(10) \equiv (\alpha e_0 - \beta e_1) \\ 11 & \mapsto & a_3(11) \equiv (\alpha e_1 - \beta e_0) \end{array}$$

Representing matrices by maze graphes

As an example, we consider the three basic matrices:

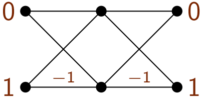
$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

It correspond to the following three graphes respectively.

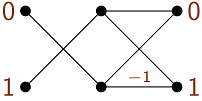


Product of matrices:

$H \cdot H$:

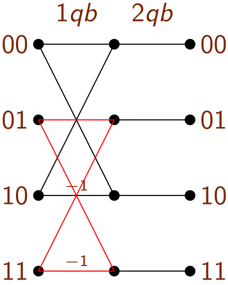


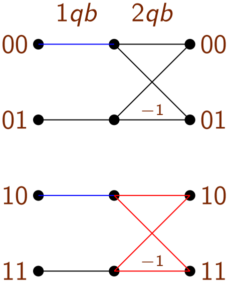
$H \cdot X$:



Tensor product of matrices: $H \otimes I$ and $I \otimes H$

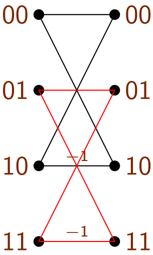
$$H \otimes I = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad I \otimes H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

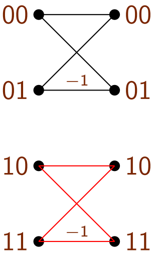




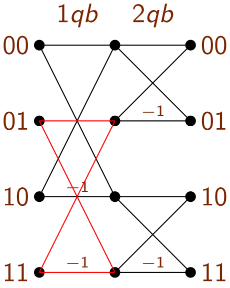
Tensor product of matrices: $H \otimes I$ and $I \otimes H$

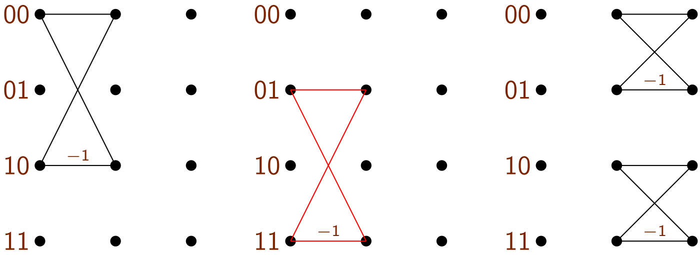
$$H \otimes I = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad I \otimes H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$





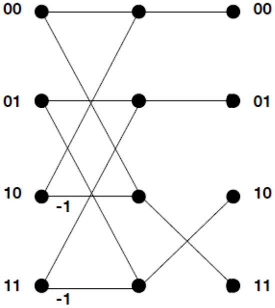
Tensor product of Hadamard matrices: $H \otimes H = (H \otimes I)(I \otimes H)$





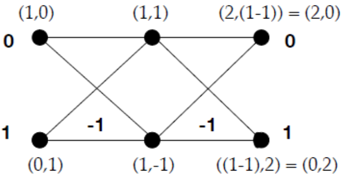
Quantum mazes:

Maze for Hadamard on qubit 1 followed by *CNOT* on 1 and 2:



$$U = (H \otimes I)(CNOT)$$

Maze for two consecutive Hadamard gates:



$$He_0 = \frac{1}{\sqrt{2}}(e_0 + e_1), He_1 = \frac{1}{\sqrt{2}}(e_0 - e_1)$$

$$H \cdot (He_0) = \frac{1}{\sqrt{2}}H(e_0 + e_1) = \frac{1}{2}(2e_0) = \frac{1}{2}(2,0)^T$$

$$H \cdot (He_1) = \frac{1}{\sqrt{2}}H(e_0 - e_1) = \frac{1}{2}(2e_1) = \frac{1}{2}(0,2)^T$$

Outline

7. Phil's Algorithm

8. Deutsch's Algorithm

9. The Deutsch-Jozsa Algorithm

8. Deutsch's Algorithm

Deutsche's algorithm operates on a Boolean function:

$$f : \{0,1\} \to \{0,1\}.$$

The function has following four forms:

$$f(x) \equiv 0;\ f(x) \equiv 1;\ f(x) = x;\ f(x) = \overline{x}.$$

The goal is to tell whether the function is a constant by performing only one evaluation of the function.

Clearly this is impossible in the classical model of computation, but the quantum model achieves this in a sense delineated below.

8.1. The Algorithm

We will present the algorithm as computing a series of vectors a_0, a_1, a_2, a_3, each of which is in the real Hilbert space $\mathbb{H}_1 \otimes \mathbb{H}_2$, where $\mathbb{H}_1$ and $\mathbb{H}_2$ are two-dimensional spaces.

We index vectors in this space by xy, where x and y are single bits.

Recall from section 4.3 that we work with its invertible extension, which we here symbolize as

$$f'(xy) = x(f(x) \oplus y).$$

Thus, the "input" to the algorithm is really the choice of f as a parameter.

By the unitary matrix U_f, we have:

$$\big|x\big\rangle \otimes \big|y\big\rangle \xrightarrow{\quad U_f \quad} U_f\big|x\big\rangle \otimes \big|y\big\rangle = \big|x\big\rangle \otimes \big|y \oplus f(x)\big\rangle.$$

The algorithm always uses the same input vector and goes as follows:

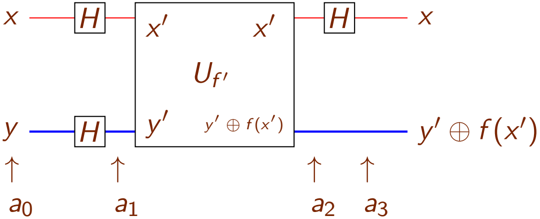
1. The initial vector is a_0 so that $a_0(01) = 1$.

2. The next vector a_1 is the result of applying the Hadamard transform on each $\mathbb{H}_i$ of the space with $i = 1, 2$ separately.

3. Then the vector a_2 is the result of applying $U_{f'}$ where

$$f'(xy) = x(f(x) \oplus y).$$

4. The final vector a_3 is the result of applying the Hadamard transform again, but this time only to $\mathbb{H}_1$.



$$U_1 = H \otimes H, U_1 = U_{f'}, U_2 = H \otimes I$$

$$U = U_3 U_2 U_1, a_3 = U a_0$$

We take initial vector $a_0 = e_{01} = \big|0\big\rangle\big|1\big\rangle = \big|0\big\rangle \otimes \big|1\big\rangle$, where

$$\big|0\big\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \big|1\big\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Note that in case f is the **identity function**, f' becomes the $Controlled - NOT$ function, and $U_{f'}$ becomes the $4 \otimes 4$ $CNOT$ matrix. Because f is the identity, we rename $U_{f'}$ to U_I. Similarly, we write U_X , U_T , and U_F for the cases f being the negation, always-true, and always-false function, respectively.

$$U_I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} (f(x) = x), \ U_X = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (f(x) = \overline{x})$$

$$U_T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} (f(x) \equiv 1), \ U_F = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (f(x) \equiv 0)$$

$$xy \to x(y \oplus f(x))$$

Note that the matrices U_T and U_F are unitary **even though the always-true and always-false functions are not reversible**.

This illustrates the quantum trick of preserving the "x" argument of these functions as the first qubit and recording $f(x)$ in terms of its effect when exclusive-or'ed with the second qubit, y.

The chain of three matrices is applied to the start vector $a_0 = e_{01}$ on the right, producing in each of the four cases the vector a_3.

$$a_3 = U_3 U_2 U_1 e_{10}$$

(1) $|0\rangle \otimes |1\rangle$;

(2) $\to \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$; (By $U_1 = H \otimes H$)

$$= \tfrac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle) + \tfrac{1}{2}(|1\rangle|0\rangle - |1\rangle|1\rangle);$$

(3) $\to U_f(\frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle) + \frac{1}{2}(|1\rangle|0\rangle - |1\rangle|1\rangle))$;

$$= \tfrac{1}{2}(|0\rangle|0 \oplus f(0)\rangle - |0\rangle|1 \oplus f(0)\rangle) \quad \text{(By } U_2 = U_{f'}\text{)}$$

$$+\tfrac{1}{2}(|1\rangle|0 \oplus f(1)\rangle - |1\rangle|1 \oplus f(1)\rangle);$$

$$= \tfrac{1}{2}(|0\rangle|0 \oplus f(0)\rangle + |1\rangle|0 \oplus f(1)\rangle)$$

$$-\tfrac{1}{2}(|0\rangle|1 \oplus f(0)\rangle + |1\rangle|1 \oplus f(1)\rangle);$$

$$= \frac{1}{\sqrt{2}}[(-1)^{f(0)}|0 > +(-1)^{f(1)}|1 >] \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$= a_2;$$

(4) $\to a_3$; (By $U_3 = H \otimes I$)

$$a_2 = \begin{cases} \pm\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \mathit{if}\ f(0) = f(1) \\ \pm\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \mathit{if}\ f(0) \neq f(1) \end{cases}$$

$$a_3 = \begin{cases} \pm|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \mathit{if}\ f(0) = f(1) \\ \pm|1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \mathit{if}\ f(0) \neq f(1) \end{cases}$$

$$= \pm|f(0) \oplus f(1)\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

By measuring first qubit, we can determinate the value of $f(0) \oplus f(1)$.

Note that $f(0) \oplus f(1) = 0$ if $f(0) = f(1)$, otherwise, $f(0) \oplus f(1) = 1$.

A measurement of a_3 will then determine whether we are in one of the two constant cases, where U_T or U_F is used, or whether we have one of the other two cases U_I or U_X, which represent the nonconstant functions f.

In classical algorithms, which need to call f twice to evaluate $f(0)$ and $f(1)$, the quantum algorithm can tell the difference with just one $U_{f'}$ oracle matrix, where again f' is the "controlled" version of f.

Measuring

We take the following two orthogonal project operators:

$$P_0 = |0\rangle\langle 0| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} (1, 0) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$P_1 = |1\rangle\langle 1| = \begin{pmatrix} 0 \\ 1 \end{pmatrix} (0, 1) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

here $P_0 + P_1 = I$.

Thus, the set $\{P_0, P_1\}$ of matrices forms a group of complete measurement matrices on **one qubit** in Hibert space of 2-dimension.

Generally, we take the following two orthogonal vectors:

$$\big| + \big\rangle \frac{1}{\sqrt{2}}(\big|0\big\rangle + \big|1\big\rangle, \quad \big| - \big\rangle \frac{1}{\sqrt{2}}(\big|0\big\rangle - \big|1\big\rangle$$

Define two orthogonal project operators:

$$P_{+} = \big| + \big\rangle\big\langle + \big|, \, P_{-} = \big| - \big\rangle\big\langle - \big|,$$

where $P_{+} + P_{-} = I$.

Thus, the set $\{P_{+}, P_{-}\}$ of matrices forms a group of complete measurement matrices on **one qubit** in Hibert space of 2-dimension.

8.2. The Analysis

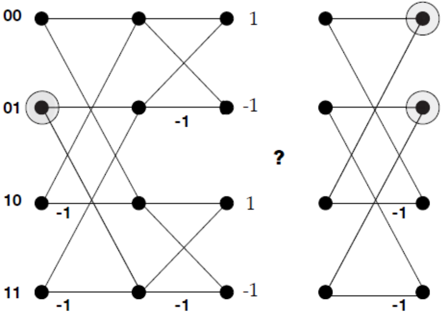
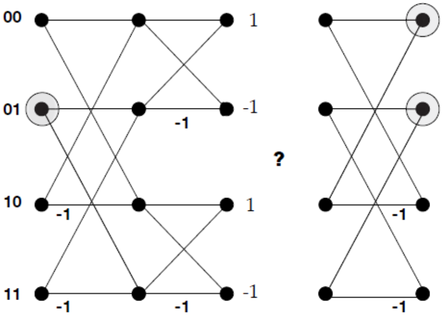


Figure 8.1. Maze for Deutsch's algorithm.

Given the input e_{01}, he enters at 01. In the figure, we see the same maze stage at left and right, which corresponds to a Hadamard gate on the first of two qubit lines.

$$\begin{aligned}
(H \otimes H)e_{01} &= (I \otimes H) \cdot (H \otimes I)e_{01} \\
&= (I \otimes H)(H \otimes I)(e_0 \otimes e_1) \\
&= \tfrac{1}{\sqrt{2}}(I \otimes H)(e_0 + e_1) \otimes e_1) \\
&= \tfrac{1}{2}(e_0 + e_1) \otimes (e_0 - e_1) \\
&= \tfrac{1}{2}(e_{00} - e_{01} + e_{10} - e_{11}) \\
&= \tfrac{1}{2}(1, -1, 1, -1)
\end{aligned}$$



One of the four matrices above is filled in the blank for $U_{f'}$ in the figure 8.1.

Each is a permutation matrix, so its four corridors will run across with no branching:

$$U_I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad U_X = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad U_F = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

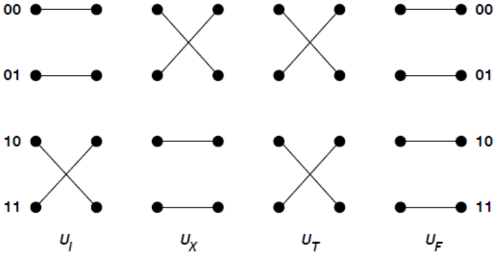


Figure 8.2. Maze stages for possible queried functions.

U_F : **makes the corridors run straight across,**

U_X : **interchanges the top two,**

U_I : **interchanges the bottom two, and**

U_T : **swaps both top two and bottom two.**

THEOREM 8.1. A measurement of the vector a_3 will return $0y$, for some y, if and only if f is a constant function. Thus, Deutsc's algorithm tells whether f is constant using just one application of $U_{f'}$.

The key of this theorem: one application of f and one measurement will tell whether f is a constant function.

To save multiplying out the 4×4 matrices for each of the four cases — a method that doesn't scale either — we use our notation indexing vectors a by $a(00), a(01), a(10), a(11)$.

The proof depends on the following lemma, in which we use binary XOR on bits to denote a number.

LEMMA 8.2. The following are true:

(1). For all xy, $a_1(xy) = \frac{1}{2}(-1)^y$.

(2). For all xy, $a_2(xy) = \frac{1}{2}(-1)^{f(x)\oplus y}$.

(3). For all xy, $|a_3(xy)|^2 = \frac{1}{8}|(-1)^{f(0)} + (-1)^{f(1)\oplus x}|^2$.

Proof. (1). It is clear that applying Hadamard gates independently yields

$$a_1(xy) = \tfrac{1}{2}\textstyle\sum_{s,t}(-1)^{x\cdot s}(-1)^{y\cdot t}a_0(st)$$

Thus, by the definition of a_0,

$$a_1(xy) = \tfrac{1}{2}(-1)^{x\cdot 0}(-1)^{y\cdot 1}a_0(01) = \tfrac{1}{2}(-1)^y.$$

(2). By definition of the matrix $U_{f'}$ it follows that

$$a_2(xy) = a_1(x(f(x) \oplus y)) = \tfrac{1}{2}(-1)^{f(x)\oplus y}.$$

Hadamard transform: $b = H_{2^n} a$

$$b(x) = \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} b(y)$$

(3) By definition of the Hadamard transform, and (2)

$$a_3(xy) = \frac{1}{\sqrt{2}} \sum_{t \in \{0,1\}} (-1)^{x \cdot t} a_2(ty)$$
$$= \frac{1}{2\sqrt{2}} \sum_{t \in \{0,1\}} (-1)^{x \cdot t} (-1)^{f(t) \oplus y}.$$

The sum is expanded, and then it is

$$\frac{1}{2\sqrt{2}}((-1)^{f(0) \oplus y} + (-1)^{x \oplus f(1) \oplus y}).$$

We can factor out the common term $(-1)^y$ to get the amplitude:

$$|a_3(xy)|^2 = \frac{1}{8} |(-1)^{f(0)} + (-1)^{x \oplus f(1)}|^2. \qquad \square$$

Proof of Theorem 8.1.

By lemma 8.2,

$$|a_3(0y)|^2 = \tfrac{1}{8}|(-1)^{f(0)} + (-1)^{f(1)}|^2.$$

If f is constant, then this expression is equal to

$$|a_3(0y)|^2 = \tfrac{1}{8} \cdot 2^2 = \tfrac{1}{2}.$$

If f is not constant, then it is equal to 0. $\square$

8.3. Superdense Coding and Teleportation

Superdense coding and teleportation applications involve a physical interpretation and realization of qubits.

Consider talk of "Alice" and "Bob" across Lake Geneva from each other.

First consider a general product state

$$\begin{aligned} c &= (a_0 e_0 + a_1 e_1) \otimes (b_0 e_0 + b_1 e_1) \\ &= a_0 b_0 e_{00} + a_0 b_1 e_{01} + a_1 b_0 e_{10} + a_1 b_1 e_{11} \end{aligned}$$

where $|a_0|^2 + |a_1|^2 = 1$ and $|b_0|^2 + |b_1|^2 = 1$.

We can regard $a = a_0 e_0 + a_1 e_1$ as a qubit wholly in the control of Alice and $b = b_0 e_0 + b_1 e_1$ as a qubit owned by Bob, with $c = a \otimes b$ standing for the joint state of the system.

For a general pure state of the form

$$d = d_{00}e_{00} + d_{01}e_{01} + d_{10}e_{10} + d_{11}e_{11}$$

with $|d_{00}|^2 + |d_{01}|^2 + |d_{10}|^2 + |d_{11}|^2 = 1$.

Alice controls the first index, which plays d_{00}, d_{01} against d_{10}, d_{11}, while Bob controls the second index, which plays the even-index entries d_{00}, d_{10} in places 0 and 2 off against the odd entries d_{01}, d_{11} in places 1 and 3.

There is one other partition that plays off two against two, the "outers" d_{00}, d_{11} versus the "inners" d_{01}, d_{10}.

This playing-off can be achieved directly by a different kind of measurement that projects onto the transformed basis whose four elements are given by $e_{00} \pm e_{11}$ and $e_{01} \pm e_{10}$, each normalized by dividing by $\sqrt{2}$.

This basis is named for John Bell:

$$B_{00} = \tfrac{1}{\sqrt{2}}(e_{00} + e_{11})$$

$$B_{01} = \tfrac{1}{\sqrt{2}}(e_{10} + e_{01})$$

$$B_{10} = \tfrac{1}{\sqrt{2}}(e_{00} - e_{11})$$

$$B_{11} = \tfrac{1}{\sqrt{2}}(e_{10} - e_{01})$$

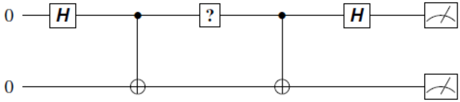
The physical realization is that after converting our usual all-zero start state e_{00} to

$$d = \frac{1}{\sqrt{2}}(e_{00} + e_{11})$$

we really can give Alice a particle representing the first coordinate and shoot Bob across the lake an entangled particle representing the second coordinate.

The experiments have demonstrated that Alice and Bob can for some time keep these particles in this joint state.

Moreover, Alice is physically able to operate further on this state by matrix operators applied only to her qubit, that is, operators of the form $U \otimes I$ where U is a 2×2 unitary matrix.



In particular, let U be one of four things:

(1) I (for 00), (2) X (for 01),

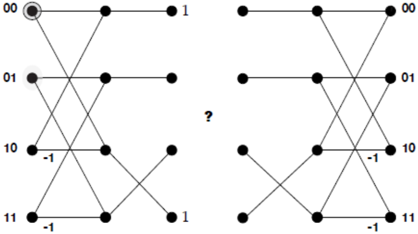
(3) Z (for 10), or (4) $iY = XZ$ (for 11)

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Thus, Alice is applying one of the Pauli matrices.

Let Alice do one of these four things and then shoot her qubit across the lake to Bob.

To show the similarity to the analysis of Deutsch's algorithm, we draw the corresponding maze diagram for the circuit with a missing stage, and the diagrams for the four possible stages Alice can insert:

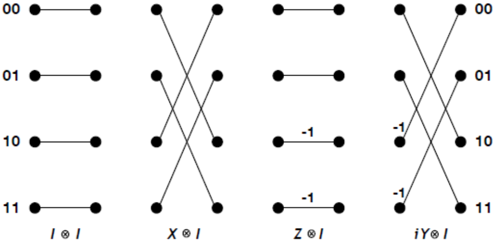


$H \otimes I \quad CNOT \quad U \otimes I \quad CNOT \quad H \otimes I$

$$\begin{array}{rcll}
e_{00} & \mapsto & \frac{1}{\sqrt{2}}(e_0 + e_1) \otimes e_0 & (By\ H \otimes I) \\
& = & \frac{1}{\sqrt{2}}(e_{00} + e_{10}) & \\
& \mapsto & \frac{1}{\sqrt{2}}(e_{00} + e_{11}) & (By\ CNOT) \\
& \mapsto & \frac{1}{\sqrt{2}}(e_{00} + e_{11}) & (By\ U \otimes I, U = I) \\
& \mapsto & \frac{1}{\sqrt{2}}(e_{00} + e_{10}) & (By\ CNOT) \\
& \mapsto & \frac{1}{2}((e_{00} + e_{10}) + (e_{00} - e_{10})) & (By\ H \otimes I) \\
& = & e_{00} &
\end{array}$$

Because the amplitude divisor is 2, this already entails that the Phils at the other exit points always cancel, but one may enjoy verifying this from the two figures. Hence, the measurement always gives the same exit point depending only on the operation Alice chose.

The main point is that Alice's four choices lead to four different results, so that Bob is able to tell what Alice did.



Why might this be surprising?

Bob has learned two bits of information as a result of the single qubit that Alice sent across the lake.

This seems to say that the one qubit carried two classical bits of information. However, there was one previous connection between them–via the intermediary who gave them the entangled qubits to begin with.

Holevo's theorem expresses the deep principle that a total transmission of n qubits can carry no more than n bits of classical information.

Thus, there must always have been some prior interaction between them or their environments to produce the entanglements.

Once they are in place, however, Alice can electively transmit information at a classically impossible two-for-one rate–at the cost of consuming entanglement resources for each pair of bits. This explains the name superdense coding.

In quantum information theory, **superdense coding is a technique used to send two bits of classical information using only one qubit**.

It is the inverse of **quantum teleportation, which sends one qubit with two classical bits.**

Both superdense coding and quantum teleportation require, and use up, entanglement between the sender (Alice) and receiver (Bob) in the form of Bell pairs.

$$B_{00} = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B)$$

are distributed to Alice and Bob.

The first subsystem, denoted by subscript A, belongs to Alice and the second, B, system to Bob.

In quantum information theory, **superdense coding is a technique used to send two bits of classical information using only one qubit**.

It is the inverse of **quantum teleportation, which sends one qubit with two classical bits.**

Both superdense coding and quantum teleportation require, and use up, entanglement between the sender (Alice) and receiver (Bob) in the form of Bell pairs.

$$B_{00} = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B)$$

are distributed to Alice and Bob.

The first subsystem, denoted by subscript A, belongs to Alice and the second, B, system to Bob.

(1) Suppose Alice wants to send the classical bits 00.

She will perform identity unitary operation on her particle. Her entangled qubit remains unchanged. The resultant tangled qubit would be

$$|B_{00}\rangle = \frac{1}{\sqrt{2}}(|0_A 0_B\rangle + |1_A 1_B\rangle).$$

(2) Suppose Alice wants to send the classical bits 01.

She will perform X unitary operation. After the application of X unitary gate the resultant entangled quantum state would be

$$|B_{01}\rangle = \frac{1}{\sqrt{2}}(|1_A 0_B\rangle + |0_A 1_B\rangle).$$

(3) Suppose Alice wants to send the classical bits 10.

She will perform Z unitary operation. After the application of Z unitary gate the resultant entangled quantum state would be

$$|B_{10}\rangle = \frac{1}{\sqrt{2}}(|0_A 0_B\rangle - |1_A 1_B\rangle).$$

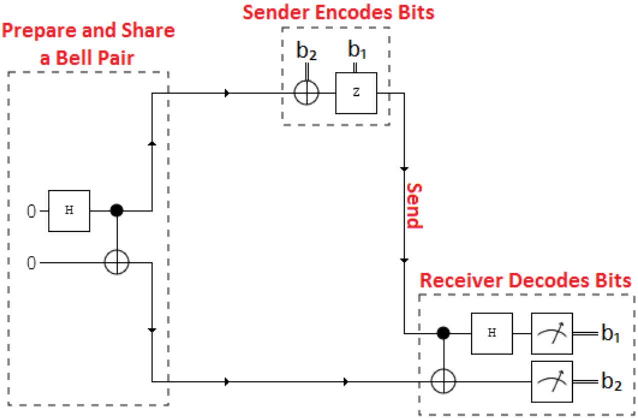
(4) Suppose Alice wants to send the classical bits 11.

She will perform XZ unitary operation. After the application of XZ unitary gate the resultant entangled quantum state would be

$$|B_{11}\rangle = \frac{1}{\sqrt{2}}(|1_A 0_B\rangle - |0_A 1_B\rangle).$$

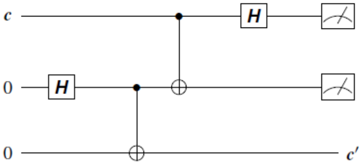
where $X, Z, I, XZ(= -iY)$ are Pauli gates. $B_{00}, B_{01}, B_{10}, B_{11}$ are called Bell states.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$


Prepare and Share a Bell Pair
Sender Encodes Bits
b2
b1
Z
0
0
H
Send
Receiver Decodes Bits
H
b1
b2

Quantum teleportation involves three qubits, two initially owned by Alice and one by Bob.

Alice and Bob share entangled qubits as before, whereas Alice's other qubit is in an arbitrary (pure) state $c = ae_0 + be_1$. Alice has no knowledge of this state, and hence cannot tell Bob how to prepare it, yet entirely by means of operations on her side of the lake, she can ensure that Bob can possess a qubit in the identical state.



$$I \otimes H \otimes I, I \otimes CNOT, CNOT \otimes I, H \otimes I \otimes I$$

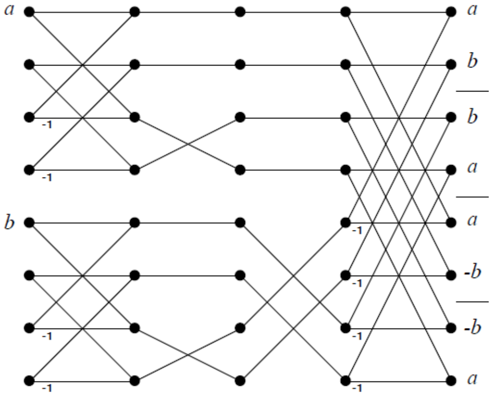
The start state is $c \otimes e_{00}$, which equals $ae_{000} + be_{100}$. After the first two gates, the state is

$$c \otimes \frac{1}{\sqrt{2}}(e_{00} + e_{11})$$

with Alice still in possession of the first coordinate of the entangled basis vectors.

The point is that the rest of the circuit involves operations by Alice alone, including the measurements, all done on her side of the lake. This is different from using a two-qubit *swap*-gate to switch the c part to Bob, which would cross the lake.

No quantum interference is involved, so a maze diagram helps visualize the results even with "arbitrary-phase Phils" lined up at the entrances for e_{000} and e_{100} as shown in the following figure.



$$\begin{aligned}
(ae_0 + be_1) \otimes e_{00} &\mapsto \tfrac{1}{\sqrt{2}}(ae_0 + be_1) \otimes (e_0 + e_1) \otimes e_0 && (I \otimes H \otimes I) \\
&= \tfrac{1}{\sqrt{2}}(ae_{000} + ae_{010} + be_{100} + be_{110}) && \\
&\mapsto \tfrac{1}{\sqrt{2}}(ae_{000} + ae_{011} + be_{100} + be_{111}) && (I \otimes CNOT) \\
&\mapsto \tfrac{1}{\sqrt{2}}(ae_{000} + ae_{011} + be_{110} + be_{101}) && (CNOT \otimes I) \\
&\mapsto \tfrac{1}{\sqrt{2}}[a(e_{000} + e_{100}) + a(e_{011} + e_{111}) + && (H \otimes I \otimes I) \\
&\qquad b(e_{010} - e_{110}) + b(e_{001} - e_{101})] && \\
&= \tfrac{1}{\sqrt{2}}(ae_{000} + be_{001} + be_{010} + ae_{011} + && \\
&\qquad +ae_{100} - be_{101} - be_{110} + ae_{111}) &&
\end{aligned}$$

Because Bob's qubit is the rightmost index, the measurement of Alice's two qubits selects one of the four pairs of values divided off by the bars at the right.

Each pair superposes to yield the value of Bob's qubit after the two measurements "collapse" Alice's part of the system.

The final step is that Alice sends two classical bits across the lake to tell Bob what results she got, that is, which quadrant was selected by nature.

The rest is in some sense the inverse of Alice's step in the superdense coding: Bob uses the two bits to select one of the Pauli operations I, X, Z, iY, respectively, and applies it to his qubit c' to restore it to Alice's original value c.

Neither Bob nor Alice is ever able to peek inside the qubit c to read the complex-number values of a and b or even get them right to more than a few uncertain bits of accuracy, amounting to at most one bit of solid information.

This is already the essence of the natural law corresponding to Holevo's theorem.

However, streams of qubits with prescribed values c can be generated, and experiments have shown that they can be received by Bob with high statistical fidelity over distances of many miles.

Outline

7. Phil's Algorithm

8. Deutsch's Algorithm

9. The Deutsch-Jozsa Algorithm

9. The Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm operates on a Boolean function:
$$f : \{0,1\}^n \to \{0,1\}.$$

The goal is to tell apart the cases where the function is constant or balanced by **performing only one evaluation of the function**. Here a function is **balanced** if it has the same number of 1's and 0's as output.

If neither case holds then the output is immaterial. Clearly this goal is impossible in the classical model of computation, even with as many as 2^{n-1} evaluations of f on Boolean arguments. However, it is possible in the quantum model with just one evaluation.

Deutsch's algorithm was important for being the first quantum algorithm, even though it only barely outperformed the classical one.

9.1. The Algorithm

We will present the algorithm as computing a series of vectors a_0, a_1, a_2, a_3, each which is in the real Hilbert space $\mathbb{H}_1 \times \mathbb{H}_2$, where $\mathbb{H}_1$ has dimension $N = 2^n$ and $\mathbb{H}_2$ has dimension 2.

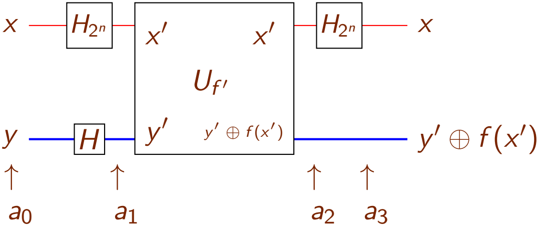
We index vectors in this space by xy, where x is n bits and y is a single bit.

1. The initial vector is a_0 so that $a_0(0^n 1) = 1$. That is, $a_0 = e_{0^n 1}$.

2. The next vector a_1 is the result of applying the Hadamard transform on each H_i of the space with $i = 1, 2$ separately.

3. Then the vector a_2 is the result of applying $U_{f'}$ where $f'(xy) = x(f(x) \oplus y)$.

4. The final vector a_3 is the result of applying the Hadamard transform again, but this time only to $\mathbb{H}_1$.



$$U_1 = H_{2^n} \otimes H, U_2 = U_{f'}, U_3 = H_{2^n} \otimes I$$

$$a_0 = 0^n 1, a_i = U_i a_{i-1} \ (i = 1, 2, 3)$$

9.2. The Analysis

THEOREM 9.1. A measurement of the vector a_3 will return $0^n y$, for some y, if and only if f is a constant function. Thus, the Deutsch-Jozsa algorithm distinguishes the cases of f being constant or balanced using only one evaluation of $U_{f'}$.

This theorem is the key: one measurement will work to tell whether f is a constant function. The proof depends on the following lemma.

LEMMA 9.2. The following are true:

(1) For all x, y, $a_1(xy) = \frac{1}{\sqrt{2N}}(-1)^y$.
(2) For all x, y, $a_2(xy) = \frac{1}{\sqrt{2N}}(-1)^{f(x)\oplus y}$.
(3) For all x, y, $|a_3(xy)|^2 = \frac{1}{2N^2}|\sum_{t\in\{0,1\}^n}(-1)^{x\cdot t}(-1)^{f(t)}|^2$.

Proof. (1). It is clear that applying the Hadamard gates independently yields

$$a_1(xy) = \frac{1}{\sqrt{2N}} \sum_{t\in\{0,1\}^n, u\in\{0,1\}} (-1)^{t\cdot x}(-1)^{u\cdot y} a_0(tu)$$

where we remind that $x \cdot t$ is the XOR-based inner product of Boolean strings, whereas $y \cdot u$ involves just single bits.

Thus, by the definition of a_0,

$$a_1(xy) = \frac{1}{\sqrt{2N}}(-1)^{x\cdot 0^n}(-1)^{1\cdot y} = \frac{1}{\sqrt{2N}}(-1)^y.$$

(2). By definition of the matrix U_g it follows that

$$a_2(xy) = a_1(x(f(x) \oplus y)) = \frac{1}{\sqrt{2N}}(-1)^{f(x)\oplus y}.$$

(3). By definition of the Hadamard transform,

$$a_3(xy) = \frac{1}{\sqrt{N}} \sum_{t\in\{0,1\}^n} (-1)^{x\cdot t} a_2(ty)$$
$$= \frac{1}{\sqrt{2N}} \sum_{t\in\{0,1\}^n} (-1)^{x\cdot t} (-1)^{f(t)\oplus y} \square$$

Proof of Theorem 9.1. By lemma 9.2,

$$|a_3(0^n y)|^2 = \frac{1}{2N^2} |\sum_{t\in\{0,1\}^n} (-1)^{x\cdot t} (-1)^{f(t)}|^2$$

If f is constant, then this expression is equal to

$$\frac{1}{2N^2} |\sum_{t\in\{0,1\}^n} (-1)^{0^n\cdot t}|^2 = \frac{1}{2}.$$

Thus, the two equivalent cases $y = 0, 1$ each have probability $\frac{1}{2}$, making it certain that the measurement yields $0^n y$.

If f is not constant, then it is equal to

$$\tfrac{1}{2N^2} | \textstyle\sum_{t \in \{0,1\}^n} (-1)^{0^n \cdot t} (-1)^{f(t)} |^2 = 0.$$

The sum $\sum_{t \in \{0,1\}^n} (-1)^{f(t)}$ is 0 because f is balanced, and so the measurement never yields $0^n y$. □