Contents

1 THE DEUTSCH-JOSZA AND GROVER'S ALGORITHMS 3

1.1 Deutsch-Josza algorithm 3

1.1.1 Matrix representation of the "oracle" 5

1.1.2 Some comments on the "phase kick-back" 6

1.1.3 Full implementation on n-qubits input 6

1.2 Grover's algorithm 8

1.2.1 Classical probabilistic algorithm 8

1.2.2 Quantum algorithm for 2-qubits problem 9

1.2.3 Quantum circuit for 2 qubits 11

1.2.4 Graphic interpretation: inversion about the mean 13

1.2.5 How to generalize? 14

1.2.6 Generalization to n qubits 15

1 THE DEUTSCH-JOSZA AND GROVER'S ALGORITHMS

1.1 Deutsch-Josza algorithm 3
1.1.1 Matrix representation of the "oracle" 5
1.1.2 Some comments on the "phase kick-back" 6
1.1.3 Full implementation on n-qubits input 6
1.2 Grover's algorithm 8
1.2.1 Classical probabilistic algorithm 8
1.2.2 Quantum algorithm for 2-qubits problem 9
1.2.3 Quantum circuit for 2 qubits 11
1.2.4 Graphic interpretation: inversion about the mean 13
1.2.5 How to generalize? 14
1.2.6 Generalization to n qubits 15

This chapter is devoted to explaining two partially connected of the (few) known quantum algorithms – which are, those algorithms presenting an effective quantum advantage with respect to classical computation. By quantum advantage, we mean precisely that performing the same task on classical hardware and quantum hardware require a (much) different number of steps, in terms of computational time.

The first one was proposed by David Deutsch and Richard Josza in 1992. This algorithm, even if of little practical use, solves the following problem exponentially faster than classical computers: we aim to find out a specific property of a given unknown binary function f acting on our (qu)bits. The first section is devoted to it.

The second one was proposed by Lov Grover four years later, and is of major practical interest: it is able to perform a database search with a certain quantum advantage. We expand in the Sec. 1.2.

1.1 DEUTSCH-JOSZA ALGORITHM

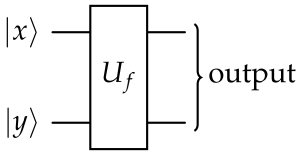
We start by a somewhat "reduced" version of the Deutsch-Josza algorithm. Let f be an unknown function acting on the computational basis,

$$f\colon \{0,1\} \to \{0,1\}$$

Of course the number of possible functions is very small. Anyhow the function acts, we define it **constant** or **balanced** if

$$\begin{aligned} f(0) \oplus f(1) = 0 \quad &\Longrightarrow \quad f(0) = f(1) \qquad f \text{ is constant} \\ f(0) \oplus f(1) = 1 \quad &\Longrightarrow \quad f(0) \neq f(1) \qquad f \text{ is balanced} \end{aligned}$$

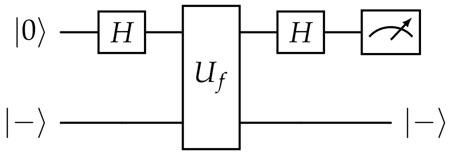
Suppose also we are provided with a 2-qubit "oracle" gate,

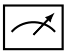


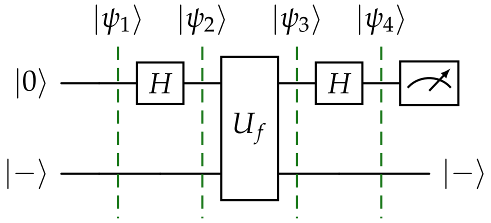
a sort of "black box" able to implement f. We do not need to know how it works physically in order to study the algorithm. Let $x, y \in \{0,1\}$. U_f acts as

$$U_f\,|x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle$$

The problem we aim to solve is: how do we know if f is constant or balanced? How many calls to the oracle, U_f, are needed? Classically, we would need to call the oracle twice: one for input 0, one for input 1, measuring the result each time. Can we do better? Take now the following circuit



where the symbol  represents measurement. To understand what happens, let us break down the circuit in steps.



1. The initial state is simply

$$|\psi_1\rangle = |0\rangle \otimes |-\rangle = |0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

2. The Hadamard gate maps the state to

$$|\psi_2\rangle = [H \otimes \mathbb{1}]\,|\psi_1\rangle = |+\rangle \otimes |-\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

3. The function is implemented:

$$\begin{aligned}|\psi_3\rangle &= U_f\,|\psi_2\rangle \\ &= \frac{|0\rangle}{\sqrt{2}} \otimes \frac{|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle}{\sqrt{2}} + \frac{|1\rangle}{\sqrt{2}} \otimes \frac{|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle}{\sqrt{2}} \\ &= (-1)^{f(0)}\frac{|0\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} + (-1)^{f(1)}\frac{|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ &= (-1)^{f(0)}\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}\,|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}\end{aligned}$$

We made use of the so-called **phase kick-back** through the second passage (see Sec. 1.1.2).

4. Finally, the Hadamard gate is applied

$$|\psi_4\rangle = [H \otimes \mathbb{1}]\,|\psi_3\rangle$$

We know that $H\,|+\rangle = |0\rangle$, $H\,|-\rangle = |1\rangle$. This implies

$$\begin{aligned} f(0) \oplus f(1) = 0 \quad &\Longrightarrow \quad |\psi_4\rangle = |0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ f(0) \oplus f(1) = 1 \quad &\Longrightarrow \quad |\psi_4\rangle = |1\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned}$$

Then, we actually need **just one call to the "oracle"** to find out if f is balanced or constant: just by measuring the first qubit, if the result is 0 the function is constant, if the result is 1 the function is balanced.

1.1.1 Matrix representation of the "oracle"

Before passing on to the n-qubits version of the algorithm, we briefly give a matrix representation of the algorithm. We make use the following Ansatz

$$U_f = |0\rangle\langle 0| \otimes U_0 + |1\rangle\langle 1| \otimes U_1$$

being U_0 and U_1 single-qubit operators. Such Ansatz makes sense: by definition

$$U_f\,|x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle$$

thus if $f = 0$ the action on the second qubit is null, and it is not if $f = 1$. We compute U_0 element-wise,

$$[U_0]_{00} = \langle 0|U_0|0\rangle = \langle 0|0 \oplus f(0)\rangle = \langle 0|f(0)\rangle = \bar{f}(0)$$

with $\bar{f}(0)$ the logical negation of $f(0)$. By the same means one gets

$$[U_0] = \begin{bmatrix} \bar{f}(0) & f(0) \\ f(0) & \bar{f}(0) \end{bmatrix} = \bar{f}(0)\mathbb{1} + f(0)X$$

and equivalently

$$[U_1] = \begin{bmatrix} \bar{f}(1) & f(1) \\ f(1) & \bar{f}(1) \end{bmatrix} = \bar{f}(1)\mathbb{1} + f(1)X$$

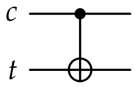
Then we can give a $2^2 \times 2^2$ matrix representation for U_f,

$$[U_f] = \begin{bmatrix} \bar{f}(0) & f(0) & & \\ f(0) & \bar{f}(0) & & \\ & & \bar{f}(1) & f(1) \\ & & f(1) & \bar{f}(1) \end{bmatrix}$$

To compute such matrix can be done for a small number of qubits; the task rapidly diverges for $n > 1$ input qubits.

1.1.2 Some comments on the "phase kick-back"

By "phase kick-back" we refer to those controlled actions having effect on the controls rather than the targets. Take for example the CNOT operator,

CNOT = 

We identify the control as the state "being read" and the target as the one "being acted upon". However, let the target be in the $|-\rangle$ state. Then, being $X|-\rangle = -|-\rangle$

$$\text{CNOT}\,|0\rangle \otimes |-\rangle = |0\rangle \otimes |-\rangle \qquad \text{CNOT}\,|1\rangle \otimes |-\rangle = -\,|1\rangle \otimes |-\rangle$$

We can interpret the result as the target remaining unchanged, and the control undergoing a π phase shift. Under this perspective we may interchange the roles of control and target. Take as well the control to be in the $|+\rangle$ state: then

$$\text{CNOT}\,|+\rangle \otimes |-\rangle = |-\rangle \otimes |-\rangle$$

Just by using a different basis we were able to "exchange roles". This is what we did in step 3 of the computation discussed in the above section: even if the second state is strictly speaking the target of U_f, the real action of the operator is a phase shift of the control.

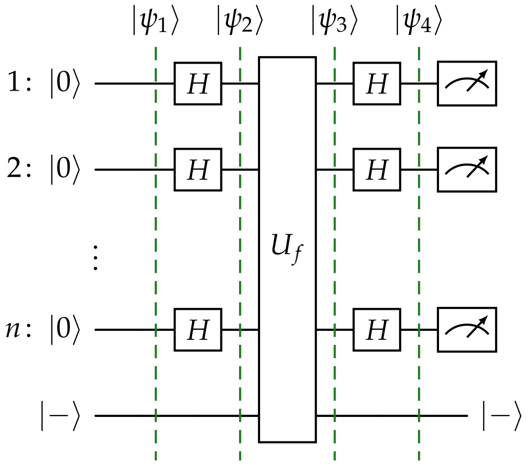
1.1.3 Full implementation on n-qubits input

Let us extend the previous result to a more interesting situation. Let f be a function of the kind

$$f\colon \{0,1\}^{\otimes n} \to \{0,1\}$$

thus taking as input a binary string of length n (any integer number up to $2^{n-1}+1$) and giving back as output a single bit. Here we assume that f is *either* constant or balanced. On n-qubits, we expand the definition of a "balanced function" as one equal to 0 on half of the domain and to 1 on the other half. We stress that we are assuming f to be this way. Expand...

The full circuit is here given by



We may proceed by exactly the same argument as in the previous section.

1. The initial state is

$$|\psi_1\rangle = [|0\rangle]^{\otimes n} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

2. Apply n parallel Hadamard gates:

$$|\psi_2\rangle = \left[H^{\otimes n} \otimes \mathbb{1}\right] |\psi_1\rangle$$

In particular:

$$H^{\otimes n} [|0\rangle]^{\otimes n} = \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right]^{\otimes n} = \frac{1}{2^{n/2}} \sum_{\mathbf{x}\in\{0,1\}^{\otimes n}} |\mathbf{x}\rangle$$

where $\mathbf{x}$ represents a string of zeros and ones of length n. Thus the state becomes

$$|\psi_2\rangle = \left[\frac{1}{2^{n/2}} \sum_{\mathbf{x}\in\{0,1\}^{\otimes n}} |\mathbf{x}\rangle\right] \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

3. The U_f gate is applied:

$$|\psi_3\rangle = \left[\frac{1}{2^{n/2}} \sum_{\mathbf{x}\in\{0,1\}^{\otimes n}} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle\right] \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

4. Finally, we apply once again n parallel Hadamard gates. Let $\mathbf{x}$ be the string

$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

with $x_i \in \{0,1\}$. Then

$$\begin{aligned} H^{\otimes n} |\mathbf{x}\rangle &= \bigotimes_{i=1}^{n} H |x_i\rangle \\ &= \bigotimes_{i=1}^{n} \frac{|0\rangle + (-1)^{x_i} |1\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle + (-1)^{x_1} |1\rangle}{\sqrt{2}} \otimes \cdots \otimes \frac{|0\rangle + (-1)^{x_n} |1\rangle}{\sqrt{2}} \\ &= \frac{1}{2^{n/2}} \sum_{\mathbf{z}\in\{0,1\}^{\otimes n}} (-1)^{\mathbf{x}\cdot\mathbf{z}} |\mathbf{z}\rangle \end{aligned}$$

since, in the last passage, the sign of the n-qubits state $|\mathbf{z}\rangle$ is controlled by how many ones are present in the string which are associated to $x_i = 1$. It follows

$$|\psi_4\rangle = \left[H^{\otimes n} \otimes \mathbb{1}\right] |\psi_3\rangle = \left[\frac{1}{2^n} \sum_{\mathbf{x}\in\{0,1\}^{\otimes n}} \sum_{\mathbf{z}\in\{0,1\}^{\otimes n}} (-1)^{f(\mathbf{x})+\mathbf{x}\cdot\mathbf{z}} |\mathbf{z}\rangle\right] \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

The last step is measurement. The amplitude for measuring the term $|z\rangle = |00\cdots 0\rangle$ is

$$\frac{1}{2^n} \sum_{\mathbf{x}\in\{0,1\}^{\otimes n}} (-1)^{f(\mathbf{x})}$$

If f is constant, then $f(\mathbf{z}) = f_0$ for any argument. Then

$$\frac{1}{2^n} \sum_{\mathbf{x} \in \{0,1\}^{\otimes n}} (-1)^{f_0} = (-1)^{f_0}$$

which implies unitary probability of measuring such state. Conversely, if f is balanced

$$\frac{1}{2^n} \sum_{\mathbf{x} \in \{0,1\}^{\otimes n}} (-1)^{f(\mathbf{x})} = 0$$

because f equals 0 on half of the domain and 1 on the other half. Then, to determine if the function is constant or balanced it suffices to measure if the system occupies the state $|00 \cdots 0\rangle$: if yes, the function is constant; if not, it is balanced. Just one call.

1.2 GROVER'S ALGORITHM

By itself, Deutsch-Josza algorithm – even if brilliant by its simplicity and its radicalism (one call instead of exponentially many) – solves a somewhat useless problem. At least some elements of it are however connected to **Grover's algorithm**, of major practical interest; this latter is able to solve the so-called **Quantum Database Search**.

The task to be solved is easily explained: we are given an **unstructured database** (a set of $N \in \mathbb{N}$ objects without any particular order); to each of these is associated one index entry x. Then we have a list of N entries (consecutive integer numbers); by knowing one entry, we can access the object it represents. Moreover, we want to locate one specific object in the database (if present), which is, extract its index entry.

We are given a function running over the database,

$$f \colon \{0, 1, \cdots, N-1\} \to \{0, 1\}$$

able to compare the object at a given entry with the target. It is not necessary to know how it internally works: all we need to know is that, given some entry, the function access the relative object and states if it is the target or not. Then $f(x)$ works as follows

$$\begin{cases} f(x) = 0 & \text{if} \quad x \neq x_0 \\ f(x) = 1 & \text{if} \quad x = x_0 \end{cases}$$

for one (and only one) specific input x_0, the index entry of the searched object.

Classically, for a database of size $N = 2^n$, we would need to run the oracle on the entirety of the database. Which is, the average number of calls would be of order $N/2 = 2^{n-1}$ – exponentially large in the binary string length. Does it get better, using a probabilistic architecture?

1.2.1 Classical probabilistic algorithm

We aim to run an algorithm able to locate x_0 in the database with probability p, arbitrarily set as $p = 2/3$. The only strategy we have is to choose randomly and compare with the target object.

In steps:

1. Guess with uniform probability some entry x_1 out of $\{0,1\}^{\otimes n}$.
2. Check if f, evaluated on the index entry extracted at the precedent step, equals 1.
   a) If yes, halt and return the index entry.
   b) If not, guess with uniform probability some new entry x_2 out of $\{0,1\}^{\otimes n} \setminus \{x_1\}$. Repeat from step 2.

At step 1, the probability of halting the program is simply

$$\text{Prob}\,\{x_1 = x_0\} = \frac{1}{2^n}$$

Each repetition of step 2 increments by the same amount the probability of finding the correct answer: after k repetitions of step 2, say, that is

$$\text{Prob}\left\{x_0 \in \{x_1, x_2, \cdots, x_k\}\right\} = \frac{k+1}{2^n} \overset{!}{=} \frac{2}{3}$$

(the symbol $\overset{!}{=}$ represents imposition). Then the optimal number of calls to f is simply

$$k^\star = \left\lceil \frac{2^{n+1}}{3} - 1 \right\rceil$$

exponential in n. Can we improve, using Quantum Mechanics?

1.2.2 Quantum algorithm for 2-qubits problem

Let us start with just 2 qubits. Then we are dealing with a 4-objects database, with entries

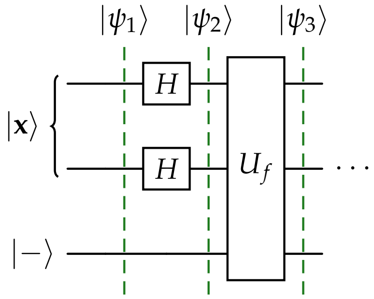
$$00 \qquad 01 \qquad 10 \qquad 11$$

Add one qubit and define the "oracle gate" U_f as in Sec. 1.1,

$$U_f\,|\mathbf{x}\rangle \otimes |y\rangle = |\mathbf{x}\rangle \otimes |y \oplus f(\mathbf{x})\rangle$$

being $|\mathbf{x}\rangle$ a 2-qubits state in the computational basis and $y \in \{0,1\}$. Suppose one of the four entries, say $\mathbf{x}_0$, points to the desired object. Then U_f acts as identity over all the states $|\mathbf{x} \neq \mathbf{x}_0\rangle$, and negates y when the index entry is correct.

As we have already seen in Sec. 1.1.2, by using **phase kick-back** we may leave the additional qubit in an unchanged superposition just by setting it in an appropriate initial superposition. As a circuit,



As we already know,

$$|\psi_3\rangle = \left[\frac{1}{2^{n/2}} \sum_{\mathbf{x}\in\{0,1\}^{\otimes n}} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle\right] \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

with $n = 2$. The sign flips only on the target entry,

$$|\psi_3\rangle = \frac{1}{2}\left[-|\mathbf{x}_0\rangle + \sum_{\mathbf{x}\neq\mathbf{x}_0} |\mathbf{x}\rangle\right] \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Can we convert this sign change into a measurable amplitude change? We should look for a 3-qubits operator to apply of the form

$$D \otimes \mathbb{1}$$

with D a 2-qubits operator, such that

$$\frac{1}{2}D\left[-|\mathbf{x}_0\rangle + \sum_{\mathbf{x}\neq\mathbf{x}_0} |\mathbf{x}\rangle\right] = |\mathbf{x}_0\rangle$$

or equivalently

$$\frac{1}{2}\sum_{\mathbf{x}} D\,|\mathbf{x}\rangle = (\mathbb{1} + D)\,|\mathbf{x}_0\rangle$$

As it can be easily checked, in the computational basis D is given by

$$[D] = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$$

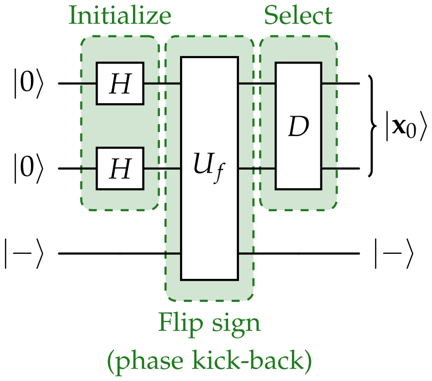
Take for example $\mathbf{x}_0 = 10$; it follows

$$|\mathbf{x}_0\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{then} \quad |\psi_3\rangle = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

so that

$$D\,|\psi_3\rangle = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = |\mathbf{x}_0\rangle$$

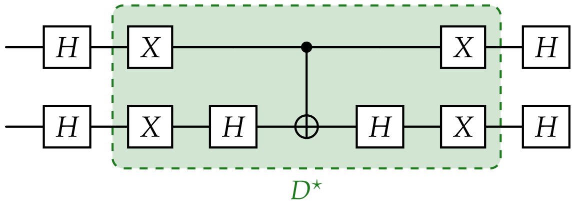
This operator performs the said conversion of phase flip to amplitude increment. Then the complete circuit is given by



The first part initializes the circuit implementing Hadamard superpositions; the second performs phase kick-back through the "oracle", thus flipping the sign of the target index entry in the superposition; the last part converts the sign flip in an amplitude change, reducing the superposition to a selected pure state - the searched index entry.

1.2.3 Quantum circuit for 2 qubits

Now, let us give a circuit representation of D. Consider the following circuit



Our claims are:

1. The sub-circuit $D^\star$ realizes

$$[D^\star] = \begin{bmatrix} 1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \end{bmatrix}$$

   in the computational basis.

2. The entire circuit realizes

$$(H \otimes H) D^\star (H \otimes H) = D$$

Let us now prove those statements.

First statement. Consider the circuit for $D^\star$. In terms of operators,

$$D^\star = (X \otimes X)(\mathbb{1} \otimes H)\,\mathrm{CNOT}(\mathbb{1} \otimes H)(X \otimes X)$$

The CNOT operator may as well be written as

$$\begin{aligned} \mathrm{CNOT} &= |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes X \\ &= \frac{1}{2}[\mathbb{1} + Z] \otimes \mathbb{1} + \frac{1}{2}[\mathbb{1} - Z] \otimes X \\ &= \frac{1}{2}[\mathbb{1} \otimes \mathbb{1} + Z \otimes \mathbb{1} + \mathbb{1} \otimes X - Z \otimes X] \end{aligned}$$

Then, using the identities $H^2 = \mathbb{1}$ and $HXH = Z$,

$$(\mathbb{1} \otimes H)\,\mathrm{CNOT}(\mathbb{1} \otimes H) = \frac{1}{2}[\mathbb{1} \otimes \mathbb{1} + Z \otimes \mathbb{1} + \mathbb{1} \otimes Z - Z \otimes Z]$$

Finally, by the identities $X^2 = \mathbb{1}$ and $XZX = -Z$

$$D^\star = \frac{1}{2}[\mathbb{1} \otimes \mathbb{1} - Z \otimes \mathbb{1} - \mathbb{1} \otimes Z - Z \otimes Z]$$

Sum and subtract one identity inside the brakets,

$$D^\star = \mathbb{1} \otimes \mathbb{1} - \frac{1}{2} [\mathbb{1} \otimes \mathbb{1} + Z \otimes \mathbb{1} + \mathbb{1} \otimes Z + Z \otimes Z]$$

Since:

$$[\mathbb{1} \otimes \mathbb{1}] = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \qquad [Z \otimes \mathbb{1}] = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & -1 \end{bmatrix}$$

$$[Z \otimes Z] = \begin{bmatrix} 1 & & & \\ & -1 & & \\ & & -1 & \\ & & & 1 \end{bmatrix} \qquad [\mathbb{1} \otimes Z] = \begin{bmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & -1 \end{bmatrix}$$

then, evidently

$$[D^\star] = \begin{bmatrix} -1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

It coincides with the desired matrix apart from an overall sign (π phase). In the following step we compose this operator with parallel Hadamards, and the global operator is composed with an identity over the third qubit. We can simply ignore this phase difference. □

Second statement. It is pretty much straightforward, from here, to compute the operator $(H \otimes H)D^\star(H \otimes H)$ using $HZH = X$,

$$(H \otimes H)D^\star(H \otimes H) = \mathbb{1} \otimes \mathbb{1} - \frac{1}{2} [\mathbb{1} \otimes \mathbb{1} + X \otimes \mathbb{1} + \mathbb{1} \otimes X + X \otimes X]$$

Since:

$$[\mathbb{1} \otimes \mathbb{1}] = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \qquad [X \otimes \mathbb{1}] = \begin{bmatrix} & & 1 & \\ & & & 1 \\ 1 & & & \\ & 1 & & \end{bmatrix}$$

$$[X \otimes X] = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & 1 & & \\ 1 & & & \end{bmatrix} \qquad [\mathbb{1} \otimes X] = \begin{bmatrix} & 1 & & \\ 1 & & & \\ & & & 1 \\ & & 1 & \end{bmatrix}$$

then the circuit has matrix representation

$$-\frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$$

which is precisely the one we stated for D, apart from the already discussed sign flip. Then the circuit implements the desired operator. □

The second part of the proof could have been taken out as well by recognizing in the circuit the simplifications $HXH = Z$ and $HXXH = \mathbb{1}$.

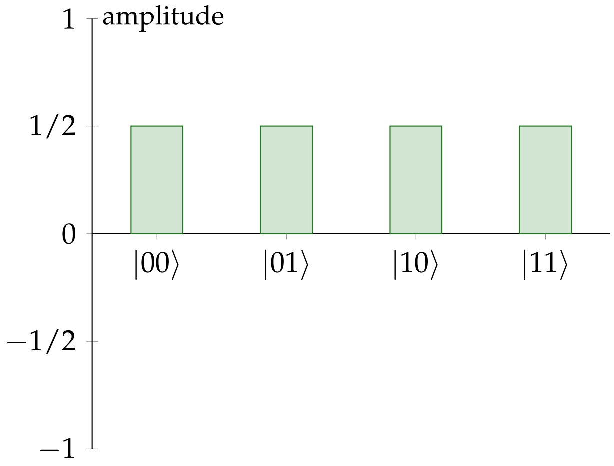


Figure 1.1: Weight distribution for the state $|\phi_2\rangle$.

1.2.4 Graphic interpretation: inversion about the mean

As we shall see in the next sections, the simplest and most useful way to think about the algorithm is via a graphic interpretation. Consider the initialized state $|\psi_2\rangle$,

$$|\psi_2\rangle = \left[\frac{1}{2^{n/2}} \sum_{\mathbf{x}\in\{0,1\}^{\otimes n}} |\mathbf{x}\rangle\right] \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |\phi_2\rangle \otimes |-\rangle$$

with

$$|\phi_2\rangle = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$

The amplitudes are all real. We can give this state a sort "weight-representation" as in Fig. 1.1. Let us see what the algorithm does on this distribution. First, it flips the sign of the target index entry (say $\mathbf{x}_0 = 10$), as in Fig. 1.2a,

$$|\phi_3\rangle = \frac{|00\rangle + |01\rangle - |10\rangle + |11\rangle}{2}$$

Then the D operation is performed (just over the subspace of $|\phi_2\rangle$), and we end up with the distribution of Fig. 1.2b,

$$|\phi_4\rangle = |10\rangle$$

What kind of mathematical operation gives the same result? In Fig. 1.2a the average amplitude is drawn as a dashed line,

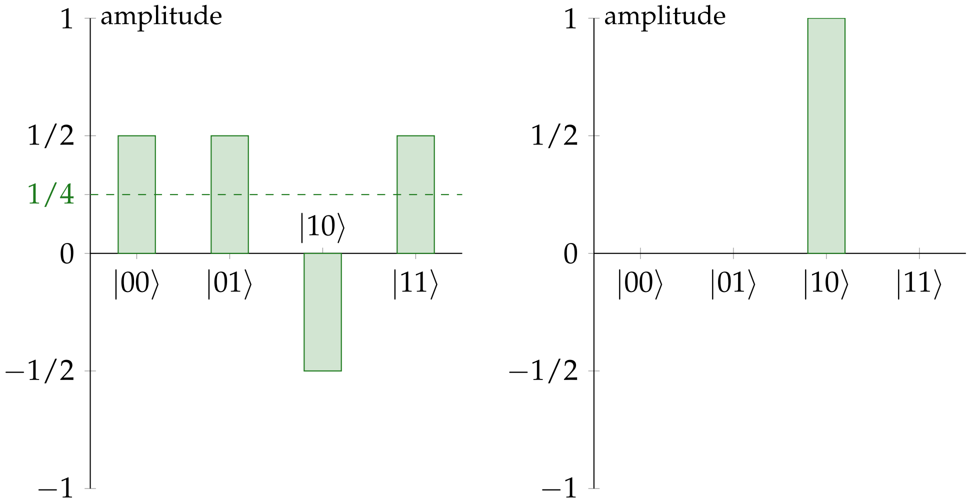
$$\alpha \equiv \langle \alpha_{\mathbf{x}} \rangle = \frac{1}{4}\left[\frac{1}{2} + \frac{1}{2} - \frac{1}{2} + \frac{1}{2}\right] = \frac{1}{4}$$

with $\alpha_{\mathbf{x}} \equiv \langle \mathbf{x}|\psi_3\rangle$. To invert the amplitude $\alpha_{\mathbf{x}}$ about the mean means to perform

$$\alpha_{\mathbf{x}} \to \alpha - (\alpha_{\mathbf{x}} - \alpha) = 2\alpha - \alpha_{\mathbf{x}}$$

Which is the correct operation indeed:

$$\frac{1}{2} \to 0 \qquad -\frac{1}{2} \to 1$$



(a) Weight distribution for the state $|\phi_3\rangle$.

(b) Weight distribution for the state $|\phi_4\rangle$.

Figure 1.2: The weight distributions for the states $|\psi_3\rangle$ (Fig. 1.2a, where the dashed line represents the average amplitude) and $|\psi_4\rangle$ (Fig. 1.2b).

Evidently inversion about the mean preserves the mean itself,

$$\langle 2\alpha - \alpha_{\mathbf{x}} \rangle = 2\alpha - \alpha = \alpha$$

So we can interpret the gate D as performing **inversion about the mean**, at least for 2 qubits. We gained quantum advantage: the oracle was called just once, instead of (on average) twice.

1.2.5 How to generalize?

From last section we learned that, to efficiently search an index entry into a 2-qubits unstructured database, we must first initialized an equally balanced superposition of computational states, then perform a phase kick-back via the oracle and finally an inversion about the mean of the final weight distribution. The question now is: can we scale up this procedure?

Take for example a 3-qubits system (a database of 8 entries). With identical notation as in the above section,

$$|\phi_2\rangle = \frac{|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle}{2\sqrt{2}}$$

Let $\mathbf{x}_0 = 010$. After phase kick-back,

$$|\phi_3\rangle = \frac{|000\rangle + |001\rangle - |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle}{2\sqrt{2}}$$

For the above distribution, the average amplitude is $3/8\sqrt{2}$; then, inverting about the mean

$$\mathbf{x} \neq 010\colon \frac{1}{2\sqrt{2}} \to \frac{1}{4\sqrt{2}} \qquad 010\colon -\frac{1}{2\sqrt{2}} \to \frac{5}{4\sqrt{2}}$$

We got close: the amplitude of the searched index entry has actually been magnified, but did not saturate. What if we repeat?

$$\mathbf{x} \neq 010\colon \frac{1}{4\sqrt{2}} \qquad 010\colon \frac{5}{4\sqrt{2}} \to -\frac{5}{4\sqrt{2}}$$

The new average is

$$\frac{1}{8}\left(\frac{7}{4\sqrt{2}}-\frac{5}{4\sqrt{2}}\right)=\frac{1}{16\sqrt{2}}$$

then

$$\mathbf{x}\neq 010\colon\ \frac{1}{4\sqrt{2}}\to -\frac{1}{8\sqrt{2}} \qquad 010\colon\ -\frac{5}{4\sqrt{2}}\to \frac{11}{8\sqrt{2}}$$

We magnified the amplitude of the most probable state, the searched one: $11/8 > 5/4$.

Repetition takes us closer to saturation. This is the core concept of Grover's algorithm: an optimal number of iterations of the process exists such that the amplitude of the searched index entry in the balanced superposition gets magnified above a certain desired level.

1.2.6 Generalization to n qubits

Finally, we discuss Grover's algorithm in its entirety: we start by the solution, and later connect it with the above sections. Let $|e\rangle$ be the equally balanced superposition in computational basis,

$$|e\rangle \equiv \frac{1}{2^{n/2}}\sum_{\mathbf{x}\in\{0,1\}^{\otimes n}}|\mathbf{x}\rangle$$

and let $\mathbf{Z}_0$ be the "extension" of the Pauli Z to n qubits,

$$\mathbf{Z}_0\,|00\cdots 0\rangle = |00\cdots 0\rangle \qquad \mathbf{Z}_0\,|\mathbf{x}\rangle = -\,|\mathbf{x}\rangle \quad \text{for} \quad \mathbf{x}\neq 00\cdots 0$$

It acts as identity over the empty string and flips sign of all other entries. Its Hadamard transform is

$$\mathbf{Z}_e \equiv H^{\otimes n}\mathbf{Z}_0 H^{\otimes n}$$

and, as can be easily guessed, acts as identity over $|e\rangle$,

$$\begin{aligned} H^{\otimes n}\mathbf{Z}_0 H^{\otimes n}\,|e\rangle &= H^{\otimes n}\mathbf{Z}_0 H^{\otimes n}H^{\otimes n}\,|00\cdots 0\rangle \\ &= H^{\otimes n}\mathbf{Z}_0\,|00\cdots 0\rangle \\ &= H^{\otimes n}\,|00\cdots 0\rangle = |e\rangle \end{aligned}$$

and flips the sign of other states $|\psi\rangle \neq |e\rangle$: states of this kind can be written as the Hadamard transform of a non-empty string $|\mathbf{x}\rangle$ with $\mathbf{x}\neq 00\cdots 0$,

$$\begin{aligned} H^{\otimes n}\mathbf{Z}_0 H^{\otimes n}\,|\psi\rangle &= H^{\otimes n}\mathbf{Z}_0 H^{\otimes n}H^{\otimes n}\,|\mathbf{x}\rangle \\ &= H^{\otimes n}\mathbf{Z}_0\,|\mathbf{x}\rangle \\ &= -H^{\otimes n}\,|\mathbf{x}\rangle = -\,|\psi\rangle \end{aligned}$$

In summary:

$$\mathbf{Z}_e\,|e\rangle = |e\rangle \qquad \mathbf{Z}_e\,|\mathbf{x}\rangle = -\,|\psi\rangle \quad \text{for} \quad |\psi\rangle \neq |e\rangle$$