Notes on an Introduction to Quantum Computing for Computer Scientists

May 2024

William Bombardelli

Introduction to Quantum Computing for Computer Scientists

The notes below might serve as an introductory text to quantum computing for computer scientist or IT professionals, in general. More than anything, these notes helped the author understanding the basics of the topic and do not have much care with form rigidity or completeness.

Introduction

In classical computing, we have a *bit*, which can be either 0 or 1. A system of two bits, can be in either one of four possible states, $00, 01, 10, 11$.
In quantum computing, we have a *qbit*, which can be either, $|0\rangle$, $|1\rangle$, or something between $|0\rangle$ and $|1\rangle$.
More precisely,

$$|0\rangle \text{ is a vector } \begin{pmatrix}1\\0\end{pmatrix} \text{ and } |1\rangle \text{ is a vector } \begin{pmatrix}0\\1\end{pmatrix} \text{ and a qbit is, in general, } \begin{pmatrix}v_0\\v_1\end{pmatrix} \text{ with } v_0^2+v_1^2=1$$

Think of the components v_0 and v_1 as the values that give the probability of that qbit being actually a 0 or a 1 in the classical world, respectively. These values are called amplitudes and are complex numbers. However, we will use only its real parts here.
Other famous qbits are

$$|+\rangle = \begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix} \text{ and } \ |-\rangle = \begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}$$

which, when measured, have both 50% chance of being either 0 or 1 when measured.
A system with two qbits can be in either one of more than just four possible states

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \quad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \quad |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \quad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

it can be in any superstate

$$\begin{pmatrix}v_{00}\\v_{01}\\v_{10}\\v_{11}\end{pmatrix} \text{ with } v_{00}^2+v_{01}^2+v_{10}^2+v_{11}^2=1 \text{ . For example, } \begin{pmatrix}1/2\\1/2\\1/2\\1/2\end{pmatrix}$$

This already gives us some whisper of the power of qbits. They can encode more states than classical bits. Beware however, that they don't offer exponential growth of states nor is a qbit in more than one state at a time. It is always in a very precise state. In other words, a quantum computer is not a non-deterministic machine.

Finally, notice that $|00\rangle$ is the tensor product of $|0\rangle$ and $|0\rangle$. That is,

$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \text{ and, in general, } \begin{pmatrix}v_{00}\\v_{01}\\v_{10}\\v_{11}\end{pmatrix} = \begin{pmatrix}w_0 * u_0\\w_0 * u_1\\w_1 * u_0\\w_1 * u_1\end{pmatrix} = \begin{pmatrix}w_0\\w_1\end{pmatrix} \otimes \begin{pmatrix}u_0\\u_1\end{pmatrix}$$

For example,

$$|{+}{+}\rangle = \begin{pmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{pmatrix} \otimes \begin{pmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{pmatrix} = \begin{pmatrix}1/2\\1/2\\1/2\\1/2\end{pmatrix} \text{ and } |{-}{-}\rangle = \begin{pmatrix}\frac{1}{\sqrt{2}}\\-\frac{1}{\sqrt{2}}\end{pmatrix} \otimes \begin{pmatrix}\frac{1}{\sqrt{2}}\\-\frac{1}{\sqrt{2}}\end{pmatrix} = \begin{pmatrix}1/2\\-1/2\\-1/2\\1/2\end{pmatrix}$$

This will give us the tools we need for the next sections.

Operations

To compute with one qbit, let's consider the following unary operations.

Identity. The identity operation is represented by I and does not do anything with the input. That is $\mathrm{I}\,|0\rangle = |0\rangle$ and $\mathrm{I}\,|1\rangle = |1\rangle$. In general, it works as a matrix multiplication over the qbit as follows.

$$\mathrm{I}\,|x\rangle = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}x_0\\x_1\end{pmatrix} = |x\rangle$$

Flip. The flip operation is represented by X and flips the components of the input. That is $\mathrm{X}\,|0\rangle = |1\rangle$ and $\mathrm{X}\,|1\rangle = |0\rangle$. In general, it works as a matrix multiplication over the qbit as follows.

$$\mathrm{X}\,|x\rangle = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}x_0\\x_1\end{pmatrix} = \begin{pmatrix}x_1\\x_0\end{pmatrix}$$

Hadamard. The hadamard operation is represented by H and puts the input in an intermediary superstate. That is $\mathrm{H}\,|0\rangle = |+\rangle$ and $\mathrm{H}\,|1\rangle = |-\rangle$. In general, it works as a matrix multiplication over the qbit as follows.

$$\mathrm{H}\,|x\rangle = \begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}\end{pmatrix}\begin{pmatrix}x_0\\x_1\end{pmatrix}$$

To compute with two qbits, let's consider the following operation.

CNOT. The CNOT operation is represented by C and works as a controlled not on the input. That is, in $\mathrm{C}\,|xy\rangle$, x controls whether y is negated or not. If x is 1, it will negate y. Example: $\mathrm{C}\,|00\rangle = |00\rangle$, $\mathrm{C}\,|10\rangle = |11\rangle$. In general, its matrix is

$$\mathrm{C} = \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}$$

Remark. Notice that applying C to superstates has some weird results.

$$\mathrm{C}\,|{+}{+}\rangle = \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}1/2\\1/2\\1/2\\1/2\end{pmatrix} = \begin{pmatrix}1/2\\1/2\\1/2\\1/2\end{pmatrix} = |{+}{+}\rangle$$

Quantum Circuits

When it comes to writing algorithms, alternatively to the matrix calculations shown above, we can use circuit diagrams. Each operation presented in the previous section has a quantum circuit notation.

Identity	Flip	Hadamard	CNOT
$\|x\rangle$ — I — $\|x\rangle$	$\|x\rangle$ — X — X $\|x\rangle$	$\|x\rangle$ — H — H $\|x\rangle$	$\|x\rangle$ — ● — $\|x\rangle$ $\|y\rangle$ — ⊕ — $\|x \oplus y\rangle$

Table 1: Quantum circuit notation of the four basic operations

Hello World

In this section, we shall explore the hello world of quantum computing. We will write Deutsch's algorithm, one of the first quantum algorithms that outperforms its classical sibling. This algorithm solves the following problem.

> Given a function $f : \{0, 1\} \to \{0, 1\}$, tell whether f is constant or variable.

f is constant if, and only if, $f(0) = f(1)$. Otherwise, f is variable, that is, $f(0) \neq f(1)$.

The Classical Algorithm

The classical algorithm for this problem is

> if $f(0) \oplus f(1) = 0$, then f is constant. Else f is variable.

Proof. f can take one of only four possible forms.

remark	name	form
constant	$set0$	$f(x) = 0$
constant	$set1$	$f(x) = 1$
variable	id	$f(x) = x$
variable	$negation$	$f(x) = \neg x$

Table 2: All four possible forms of $f : \{0, 1\} \to \{0, 1\}$

So, in the case of f being constant, f is either $set0$ or $set1$, thus, either we have $f(0) \oplus f(1) = 0 \oplus 0 = 0$ or $f(0) \oplus f(1) = 1 \oplus 1 = 0$, which proves the positive cases.
Shall f be variable, f is either id or $negation$, thus, either we have $f(0) \oplus f(1) = 0 \oplus 1 = 1$ or $f(0) \oplus f(1) = 1 \oplus 0 = 1$, which proves all the remaining cases. □

Notice that, it takes this algorithm two calls to f to give the solution. Using quantum computation, we can build an algorithm that calls f only once to decide the solution.

The Quantum Algorithm: Deutsch's Algorithm

In order to involve f in a quantum algorithm, we will explore the quantum circuit encoding of all its four possible forms, as shown in Table 2.
Creating circuits for id and $negation$ would be fairly straightforward, however, $set0$ and $set1$ are not reversible, but *quantum computation needs to be reversible.* So we need to encode them in a reversible manner.

To do this, we create an $f' : \{0,1\}^2 \to \{0,1\}^2$ such that $f'(0,x) = (f(x), x)$. In other words, we fix the first parameter as zero and record the input x in the second component of the result. This makes $set0$ and $set1$ reversible.
The four possible forms of f' are coded as shown below.

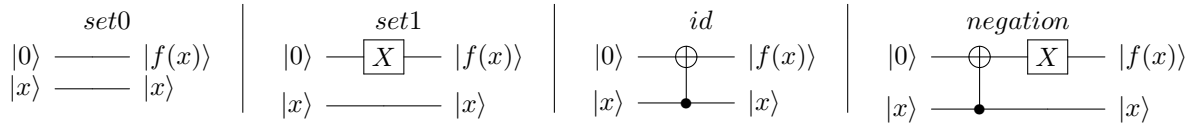
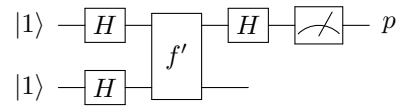
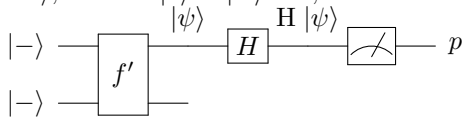


Table 3: Quantum circuits for the four forms of f'

The quantum circuit that solves our hello world problem is presented below. Mind that, as stated, we do not know f beforehand, so we have its quantum form f' as a blackbox in our circuit. We hard-code the two inputs as $|1\rangle$ and care only about the first output p, which measures 1 if, and only if, f is constant, otherwise it measures 0 (and f is variable).



Proof. In the first part of the circuit, we only do the pre-processing of putting the input $|11\rangle$ in the superstate $|{-}{-}\rangle$, since $\mathrm{H}\,|1\rangle = |-\rangle$. So, now we have,



Hence, it remains to show that $\mathrm{H}\,|\psi\rangle$ measures 1 when f' is $set0$ or $set1$, and 0 when f' is id or $negation$. These four cases are presented below.

set0 When f' is $set0$, see Table 3, $f'(|{-}{-}\rangle) = |{-}{-}\rangle$, thus $\psi = |-\rangle$ and $\mathrm{H}\,|\psi\rangle = \mathrm{H}\,|-\rangle = |1\rangle$. In which case p is 1.

set1 When f' is $set1$, see Table 3,

$$f'(|{-}{-}\rangle) = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes |-\rangle, \text{ thus } \psi = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\text{and then } \mathrm{H}\,|\psi\rangle = \mathrm{H}\,|\begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \text{ which measures 1}$$

id When f' is id, see Table 3,

$$f'(|{-}{-}\rangle) = \mathrm{C}\,|{-}{-}\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1/2 \\ -1/2 \\ -1/2 \\ 1/2 \end{pmatrix} = \begin{pmatrix} 1/2 \\ -1/2 \\ 1/2 \\ -1/2 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = |+{-}\rangle$$

$$\text{and then } \mathrm{H}\,|\psi\rangle = \mathrm{H}\,|+\rangle = |0\rangle \text{ which measures 0}$$

negation When f' is $negation$, see Table 3,

$$f'(|{-}{-}\rangle) = (X \otimes I)\,\mathrm{C}\,|{-}{-}\rangle = (\mathrm{X}\ \otimes \mathrm{I}\ )|+{-}\rangle = (\mathrm{X}\,|+\rangle \otimes \mathrm{I}\,|-\rangle)$$

$$\text{as we are interested only in the first qbit, that is } \mathrm{X}\,|+\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = |+\rangle$$

$$\text{then we have } \mathrm{H}\,|\psi\rangle = \mathrm{H}\,|+\rangle = |0\rangle \text{ which measures 0}$$

This shows all the cases and therefore that the circuit solved Deutsch's problem. Note that we made the arbitrary choice to hard-code $|0\rangle$ as the first input of the quantum circuits for f'. The algorithm above would work similarly if we arbitrated $|1\rangle$ as the first input for f'. □

Perceive that we only call f once (via f'). As much as this sounds like only a little gain, in the generalized version of this problem, the Deutsh-Josza problem, where $f : \{0,1\}^n \to \{0,1\}^n$, we can construct a similar quantum circuit that calls f only once, whereas its classical dual requires an exponential number of calls.
Furthermore, the biggest advantage of this quantum algorithm is not that *it can process things in parallel*. After all, adding parallelism to the classical algorithm would not make it defeat its quantum counterpart. Instead, the biggest gain comes from the fact that a qbit holds more than just one bit of information, it contains two (at least when using $\mathbb{R}$). That allows the first input of f' to gain knowledge about the second input and carry it out across the circuit without any cost. The way information flows through quantum circuits is key to its power.
Applications where quantum algorithms defeat classical algorithms include factoring of natural numbers for cryptography, fourier transforms and search problems. Whether graph isomorphism is one such case where quantum is better, is still unknown.
Anyhow, quantum algorithms can currently really be applied to very specific problems and do not represent a general gain of computational power. In particular, quantum computers are equivalent to turing machines as per computational power and it is not believed that they can solve NP-complete problems in polymonial time.

References

C. Moore and M. Stephan. *The Nature of Computation*. OUP Oxford, 2011.

M. Research. Quantum computing for computer scientists. URL https://www.youtube.com/watch?v=F_Riqjdh2oM.