Quantum Computation
(Lecture 6)

Luís Soares Barbosa

Universidade do Minho

HASLab
HIGH-ASSURANCE
SOFTWARE LABORATORY

INL
INTERNATIONAL IBERIAN
NANOTECHNOLOGY
LABORATORY

UNITED NATIONS
UNIVERSITY
UNU-EGOV

MSc Physics Engineering

Universidade do Minho, 2021-22

Encoding information in phases

In the Bernstein-Vazirani algorithm the key insight to find the solution w in just one application of the oracle, lies in its encoding in the relative phases between the basis states, cf

$$
\begin{aligned}
&(H^{\otimes n} \otimes I) U_f (H^{\otimes n} \otimes H) \, |0\rangle_n |1\rangle_1 \\
&\quad = (H^{\otimes n} \otimes I) U_f \left(\frac{1}{\sqrt{2^n}} \sum_{x=1}^{2^n} |x\rangle \right) \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\
&\quad = \frac{1}{\sqrt{2^n}} \left(H^{\otimes n} \sum_{x=1}^{2^n} (-1)^{f(x)} |x\rangle \right) \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\
&\quad = \frac{1}{2^n} \sum_{x=1}^{2^n} \sum_{y=1}^{2^n} (-1)^{f(x)+x \cdot y} |y\rangle \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\
&\quad = |w\rangle_n |1\rangle_1
\end{aligned}
$$

Encoding information in phases

Actually, in several quantum algorithms information is encoded in the relative phases of a quantum state.

The effect of Hadamard (once again)

$$
\begin{aligned}
H|x\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x}|1\rangle) = \frac{1}{\sqrt{2}}\sum_{y\in 2}(-1)^{xy}|y\rangle \\
H^{\otimes n}|x\rangle &= \frac{1}{\sqrt{2^n}}\sum_{y\in 2^n}(-1)^{x\cdot y}|y\rangle
\end{aligned}
$$

is to encode information about the value of x into the phases $(-1)^{x\cdot y}$ of basis states $|y\rangle$.

Encoding information in phases

Of course, as a reversible gate, it also decodes information from phases:

$$\begin{aligned} H^{\otimes n}\frac{1}{\sqrt{2^n}}\sum_{y\in 2^n}(-1)^{x\cdot y}|y\rangle &= H^{\otimes n}(H^{\otimes n}|x\rangle) \\ &= (H^{\otimes n}H^{\otimes n})|x\rangle \\ &= I|x\rangle \\ &= |x\rangle \end{aligned}$$

Encoding information in phases

In general, phases are complex numbers

$$e^{2\pi i w}$$

for any real $w \in [0, 1]$.
Of course, $H^{\otimes n}$ cannot encode/decode information over such generic phases. The general situation can be described as follows:

The phase estimation problem

Determine a good estimation of the phase parameter w given a general quantum state

$$\frac{1}{\sqrt{2^n}} \sum_{y \in 2^n} e^{2\pi i w y} |y\rangle$$

An algorithm for phase estimation

Notation

$$w = 0\,.\,x_1 x_2 \cdots$$

is written in base 2 (thus, $w = x_1 2^{-1} + x_2 2^{-2} + \cdots$); thus

$$2^k w = x_1 x_2 \cdots x_k\,.\,x_{k+1} x_{k+2} \cdots$$

and

$$\begin{aligned} e^{2\pi i (2^k w)} &= e^{2\pi i (x_1 x_2 \cdots x_k\,.\,x_{k+1} x_{k+2} \cdots)} \\ &= e^{2\pi i (x_1 x_2 \cdots x_k)}\, e^{2\pi i (0\,.\,x_{k+1} x_{k+2} \cdots)} \\ &= e^{2\pi i (0\,.\,x_{k+1} x_{k+2} \cdots)} \end{aligned}$$

because $e^{2\pi i z} = 1$ for any integer z.

Case A: 1-qubit state and $w = 0.x_1$

$$
\begin{aligned}
\frac{1}{\sqrt{2}} \sum_{y \in 2} e^{2\pi i (0.x_1) y} |y\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y \in 2} e^{2\pi i (\frac{x_1}{2}) y} |y\rangle \\
&= \frac{1}{\sqrt{2}} \sum_{y \in 2} e^{\pi i (x_1 y)} |y\rangle \\
&= \frac{1}{\sqrt{2}} \sum_{y \in 2} (-1)^{x_1 y} |y\rangle \\
&= \frac{1}{\sqrt{2}} (|0\rangle + (-1)^{x_1} |1\rangle)
\end{aligned}
$$

Clearly H will decode and retrieve x_1 because

$$
H\left(\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle)\right) = |x_1\rangle
$$

Case B: 2-qubit state and $w = 0.x_1x_2$

Observe that

$$\frac{1}{\sqrt{2^2}} \sum_{y \in 2^2} e^{2\pi i(0.x_1x_2)y} |y\rangle = \left(\frac{|0\rangle + e^{2\pi i(0.x_2)}|1\rangle}{\sqrt{2}} \right) \otimes \left(\frac{|0\rangle + e^{2\pi i(0.x_1x_2)}|1\rangle}{\sqrt{2}} \right)$$

which means that x_2, but not x_1, can be retrieved from the first qubit through an application of H.

The phase rotator

$$R_2 = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{4}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i(0.01)} \end{bmatrix}$$

where 0.01 is in base 2 (thus, equal to 2^{-2}).

Case B: 2-qubit state and $w = 0.x_1x_2$

Taking $x_2 = 1$ and applying the inverse of the phase rotator to the second qubit, yields

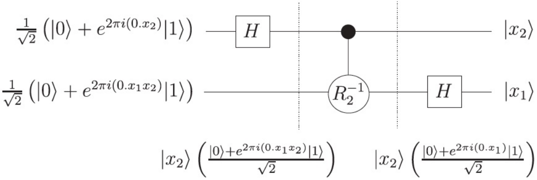
$$
\begin{aligned}
R_2^{-1}\left(\frac{|0\rangle + e^{2\pi i(0.x_1 1)}|1\rangle}{\sqrt{2}}\right) &= \begin{bmatrix} 1 & 0 \\ 0 & e^{-2\pi i(0.01)} \end{bmatrix}\left(\frac{|0\rangle + e^{2\pi i(0.x_1 1)}|1\rangle}{\sqrt{2}}\right) \\
&= \frac{|0\rangle + e^{2\pi i(0.x_1 1 - 0.01)}|1\rangle}{\sqrt{2}} \\
&= \frac{|0\rangle + e^{2\pi i(0.x_1)|1\rangle}}{\sqrt{2}}
\end{aligned}
$$

Concluding

- x_1 can now be determined by an application of H, as before.
- Moerevoer, the decision to apply R before the application of H depends on x_2 being 1 or 0, respectively.
- Thus, to find $w = 0.x_1x_2$ it is enough to apply a controlled version of R, precisely controlled by the state of the first qubit.

Case B: 2-qubit state and $w = 0.x_1x_2$

The circuit


$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.x_2)}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.x_1x_2)}|1\rangle\right)$
H
R_2^{-1}
H
$|x_2\rangle$
$|x_1\rangle$
$|x_2\rangle\left(\frac{|0\rangle+e^{2\pi i(0.x_1x_2)}|1\rangle}{\sqrt{2}}\right)$
$|x_2\rangle\left(\frac{|0\rangle+e^{2\pi i(0.x_1)}|1\rangle}{\sqrt{2}}\right)$

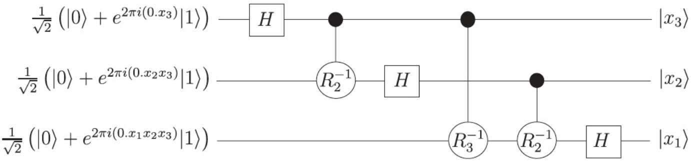
Case C: 3-qubit state and $w = 0.x_1x_2x_3$

The state is now

$$\frac{1}{\sqrt{2^3}} \sum_{y \in 2^3} e^{2\pi i (0.x_1x_2x_3)y} |y\rangle \;=$$

$$= \left(\frac{|0\rangle + e^{2\pi i (0.x_3)}|1\rangle}{\sqrt{2}} \right) \otimes \left(\frac{|0\rangle + e^{2\pi i (0.x_2x_3)}|1\rangle}{\sqrt{2}} \right) \otimes \left(\frac{|0\rangle + e^{2\pi i (0.x_1x_2x_3)}|1\rangle}{\sqrt{2}} \right)$$

In this case the third qubit has to conditionally rotate both x_2 and x_3, leading to the following circuit


$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.x_3)}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.x_2x_3)}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.x_1x_2x_3)}|1\rangle\right)$
H
R_2^{-1}
H
R_3^{-1}
R_2^{-1}
H
$|x_3\rangle$
$|x_2\rangle$
$|x_1\rangle$

Going generic

Gate R_3 in the circuit is an instance of a 1-qubit phase rotator

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}$$

whose inverse acts as

$$\begin{aligned} R_k^{-1}|0\rangle &= |0\rangle \\ R_k^{-1}|1\rangle &= e^{-2\pi i(0.0\cdots 1)}|1\rangle \end{aligned}$$

with 1 in $0.0\cdots1$ appearing in position k.

Going generic

The output state of the circuit is

$$|x_3 x_2 x_1\rangle$$

Thus, relabelling the qubits in reverse order, this provides an efficient circuit to estimate the phase (actually, to give a totally accurate estimation ...), by computing

$$\frac{1}{\sqrt{2^n}} \sum_{y \in 2^n} e^{2\pi i (\frac{x}{2^n}) y} |y\rangle \quad \rightsquigarrow \quad |x\rangle$$

Inverting ...

The inverse of the phase estimation transformation computes

$$|x\rangle \quad \rightsquigarrow \quad \frac{1}{\sqrt{2^n}} \sum_{y \in 2^n} e^{2\pi i \left(\frac{x}{2^n} \right) y} |y\rangle$$

which is obtained by taking the inverses of each gate and building the circuit in reverse order.

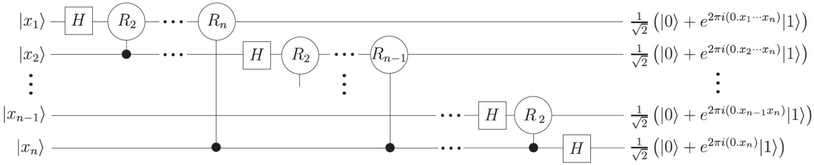
The result is formally identical to the discrete Fourier transform.

The quantum Fourier transform

QFT on basis states $|0\rangle, |1\rangle \cdots |k-1\rangle$

$$QFT_k(|x\rangle) = \frac{1}{\sqrt{k}} \sum_{y=0}^{k-1} e^{2\pi i (\frac{x}{k}) y} |y\rangle$$

The circuit


$|x_1\rangle$
$|x_2\rangle$
$|x_{n-1}\rangle$
$|x_n\rangle$
H
R_2
R_n
R_{n-1}
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.x_1\cdots x_n)}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.x_2\cdots x_n)}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.x_{n-1}x_n)}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.x_n)}|1\rangle\right)$

The quantum Fourier transform

Complexity (number of gates)

- one H plus $n-1$ conditional rotations on the first qubit
- one H plus $n-2$ conditional rotations on the second qubit
- ...

$$n+(n-1)+(n-2)+\cdots+1 \;=\; \frac{n(n-1)}{2}$$

- plus $\frac{n}{2}$ swaps (each implemented by 3 CNOT gates)

Thus

$$\frac{n(n-1)}{2}+3x\frac{n}{2} \;=\; \frac{n^2+2n}{2} \;\approx\; \mathcal{O}(n^2)$$

The quantum Fourier transform

Complexity (number of gates)

$$\frac{n(n-1)}{2} + 3x\frac{n}{2} = \frac{n^2+2n}{2} \approx \mathcal{O}(n^2)$$

which compares to the classical case for the Fast FT: $\mathcal{O}(n2^n)$

The result is impressive: the quantum version requires exponentially less operations to compute the Fourier transform than the (best) classical one.

- However, typical uses (e.g. in speech recognition) are limited by the impossibility of directly measuring the Fourier transformed amplitudes of the original state.
- This requires a subtler use of QFT in practice: the phase estimation procedure, underlying many quantum algorithms, is one of them.

Are we done?

- The circuit for QFT_k computes the QFT for k a power of 2, i.e $k = 2^n$
- The phase estimation algorithm works only when the phase is of the form $w = 0.x_1x_2 \cdots x_n$, i.e. $\frac{x}{2^n}$ for some integer x

However, it can be shown that, for an arbitrary w, the algorithm will compute x such that $\frac{x}{2^n}$ is closest to w with high probability.

The question

What is the error emerging when w is not an integer multiple of $\frac{x}{2^n}$?

Are we done?

QFT^{-1} computes some superposition

$$\sum_x \alpha_x(w)|x\rangle$$

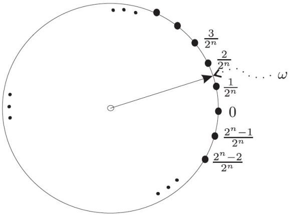
which represents the values of x that once measured give a good estimate of w, outputing x with probability $\| \alpha_x(w) \|^2$.

This output x corresponds to an estimate

$$\tilde{w} = \frac{x}{2^n}$$

Are we done?

Consider w an integer not multiple of $\frac{1}{2^n}$, and let $\hat{w}$ be the nearest integer multiple of $\frac{1}{2^n}$ to w, i.e. $\hat{w} = \frac{\hat{x}}{2^n}$ is the closest number of this form to w.



Theorem

The phase estimation algorithm returns $\hat{x}$ with probability at least $\frac{4}{\pi^2}$. i.e. the algorithm outputs an estimate $\hat{x}$ with the given probability such that

$$\left|\frac{\hat{x}}{2^n} - w\right| \leq \frac{1}{2^{n+1}}$$

Are we done?

Theorem

$$\text{If} \quad \frac{x}{2^n} \leq w \leq \frac{x+1}{2^n}$$

The phase estimation algorithm returns either x or $x+1$ with probability at least $\frac{8}{\pi^2}$ i.e. the algorithm outputs an estimate $\hat{x}$ with the given probability such that

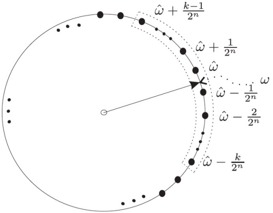
$$\left| \frac{\hat{x}}{2^n} - w \right| = \frac{1}{2^n}$$

The reverse question

How many qubits are required to get w accurate to n bits, with a probability p below a certain level?

Actually, the crucial choice is the value of n (number of qubits used) to ensure the estimation is close enough.

For $p = 1 - \frac{1}{2(k-1)}$, the algorithm returns one of the 2k closest integer multiples of $\frac{1}{2^n}$, i.e.


$\hat{\omega} + \frac{k-1}{2^n}$
$\hat{\omega} + \frac{1}{2^n}$
$\hat{\omega}$
ω
$\hat{\omega} - \frac{1}{2^n}$
$\hat{\omega} - \frac{2}{2^n}$
$\hat{\omega} - \frac{k}{2^n}$

which means that $|w - \hat{w}| \leq \frac{k}{2^n}$.

The reverse question

Thus, to estimate $\hat{w}$ such that $|w - \hat{w}| \leq \frac{1}{2^r}$. with probability at least

$$1 - \frac{1}{2^m}$$

the maximum number of qubits required is

$$n \; = \; r + m + 1$$

- In practice a much smaller error is obtained: for example, with probability at least $\frac{8}{\pi^2}$, the error will be at most

$$\frac{1}{2^{r+m}}$$

Exercises

Recall the definition of QFT on K basis states:

$$QFT_K(|x\rangle) \; = \; \frac{1}{\sqrt{K}} \sum_{y=0}^{K-1} e^{2\pi i (\frac{x}{K}) y} |y\rangle$$

Exercise 1

Compute $QFT_K(|00\cdots 0\rangle)$.

Exercise 2

Verify the following equality, used in the slides but not proved.

$$QFT_K(|x_1 \cdots x_n\rangle) \; = \; \left(\frac{|0\rangle + e^{2\pi i (0.x_n)}|1\rangle}{\sqrt{2}} \right) \otimes \left(\frac{|0\rangle + e^{2\pi i (0.x_n x_{n-1})}|1\rangle}{\sqrt{2}} \right) \cdots \otimes \cdots \left(\frac{|0\rangle + e^{2\pi i (0.x_1 x_2 \cdots x_n)}|1\rangle}{\sqrt{2}} \right)$$

Exercises

Hint to Exercise 2: The case of QFT_4 applied to $|x\rangle = |x_1 x_2\rangle$

$$
\begin{aligned}
QFT_4(|x\rangle) &= \frac{1}{2} \sum_{y=0}^{3} e^{2\pi i x y 2^{-2}} |y\rangle \\
&= \frac{1}{2} \sum_{y_1, y_2 = 0}^{1} e^{2\pi i x (y_1 2^{-1} + y_2 2^{-2})} \; |y_1 y_2\rangle \\
&= \frac{1}{2} \sum_{y_1, y_2 = 0}^{1} (e^{2\pi i x y_1 2^{-1}} |y_1\rangle \otimes e^{2\pi i x y_2 2^{-2}} |y_2\rangle) \\
&= \frac{1}{2} \sum_{y_1 = 0}^{1} (e^{2\pi i x y_1 2^{-1}} |y_1\rangle \otimes \sum_{y_2 = 0}^{1} e^{2\pi i x y_2 2^{-2}} |y_2\rangle) \\
&= \frac{(|0\rangle + e^{2\pi i x 2^{-1} |1\rangle})}{\sqrt{2}} \otimes \frac{(|0\rangle + e^{2\pi i x 2^{-2} |2\rangle})}{\sqrt{2}} \\
&= \frac{(|0\rangle + e^{2\pi i (x_1 . x_2) |1\rangle})}{\sqrt{2}} \otimes \frac{(|0\rangle + e^{2\pi i (0 . x_1 x_2) |2\rangle})}{\sqrt{2}} \\
&= \frac{(|0\rangle + e^{2\pi i (0 . x_2) |1\rangle})}{\sqrt{2}} \otimes \frac{(|0\rangle + e^{2\pi i (0 . x_1 x_2) |2\rangle})}{\sqrt{2}}
\end{aligned}
$$

Exercises

Exercise 3

One way to show that QFT is a unitary gate is by building a unitary quantum circuit for its computation. Give an alternative, direct proof that the linear transformation defined above is unitary. Note that a somehow indirect, but easy way to show an operator is unitary, is to recall that unitarian operators preserve internal products:

$$(U|v\rangle, U|u\rangle) \;=\; \langle v|U^{\dagger}U|u\rangle \;=\; \langle v, u\rangle$$

Apply this strategy to answer the exercise.

Exercise 4

Reproduce the circuit for QFT_4 and QFT_8, and compute in detail he corresponding matrices.