Cryptography & Graphs

Research Group at the University of Lleida

Quantum algorithms

SXD

IMTech & BSC

29/6/2023

Registers of q-bits

Composite systems

The algebra $\mathbb{H}^{(n)}$ of a q-register

Basis of $\mathbb{H}^{(n)}$ derived from $\{1, \mathbf{j}\}$ (or $\{|0\rangle, |1\rangle\}$)

Split elements and the Segre conditions

The state space $\Sigma^{(n)} = \mathbb{P}\mathbb{H}^{(n)}$

Split and entangled states

If $H_1, \ldots, H_n$ are the hermitian spaces of n quantum systems, the hermitian space $H_1 \otimes \cdots \otimes H_n$ defines the *composition* of those systems.[1]

The hermitian scalar product of the composite system is determined by the following rule:

$$\langle x_1 \otimes \cdots \otimes x_n | x_1' \otimes \cdots \otimes x_n' \rangle = \langle x_1 | x_1' \rangle \cdots \langle x_n | x_n' \rangle.$$

[1] We refer to [1] for a justification of this postulate.

The hermitian space of n qbits, considered as a single quantum system, is $\mathbb{H}^{(n)} = \mathbb{H}^{\otimes n}$, where the n factors $\mathbb{H}$ refer to the ordered array formed by the q-bits.

This description has an *important feature that is not present in the conventional treatment* of q-registers: $\mathbb{H}^{(n)}$ is a *unital associative* $\mathbb{C}$*-algebra*. Its structure is determined by $\mathbb{C}$-multilinearity and the rule

$$(\mathfrak{q}_1 \otimes \cdots \otimes \mathfrak{q}_n) \cdot (\mathfrak{q}'_1 \otimes \cdots \otimes \mathfrak{q}'_n) = (\mathfrak{q}_1\mathfrak{q}'_1) \otimes \cdots \otimes (\mathfrak{q}_n\mathfrak{q}'_n). \tag{1}$$

The Hermitian scalar product of $\mathbb{H}^{(n)}$ is determined by the rule

$$\langle \mathfrak{q}_1 \otimes \cdots \otimes \mathfrak{q}_n | \mathfrak{q}'_1 \otimes \cdots \otimes \mathfrak{q}'_n \rangle = \langle \mathfrak{q}_1 | \mathfrak{q}'_1 \rangle \cdots \langle \mathfrak{q}_n | \mathfrak{q}'_n \rangle \tag{2}$$

and $\bar{\mathbb{C}}/\mathbb{C}$-multilinearity. We note that $\mathfrak{q}_1 \otimes \cdots \otimes \mathfrak{q}_n$ *and* $\mathfrak{q}'_1 \otimes \cdots \otimes \mathfrak{q}'_n$ *are orthogonal if and only if* $\mathfrak{q}_k$ *and* $\mathfrak{q}'_k$, for some $k \in 1..n$, are orthogonal. Note also that

$$|\mathfrak{q}_1 \otimes \cdots \otimes \mathfrak{q}_n|^2 = |\mathfrak{q}_1|^2 \cdots |\mathfrak{q}_n|^2. \tag{3}$$

Let $B = \{0, 1\}$. For each $\nu = (\nu_1, \ldots, \nu_n) \in B^n$, set

$$\mathsf{j}^\nu = \mathsf{j}^{\nu_1} \otimes \cdots \otimes \mathsf{j}^{\nu_n} \in \mathbb{H}^{(n)}.$$

Then $\{\mathsf{j}^\nu \mid \nu \in B^n\}$ *is an orthonormal basis* of $\mathbb{H}^{(n)}$ and hence a general element of $\mathbb{H}^{(n)}$ has the form

$$\xi = \sum_{\nu \in B^n} \xi_\nu \mathsf{j}^\nu,\ \ \xi_\nu \in \mathbb{C}.$$

We have $\mathsf{j}^\nu \cdot \mathsf{j}^{\nu'} = \epsilon(\nu, \nu')\mathsf{j}^{\nu+\nu'}$, where $\epsilon(\nu, \nu')$ is the parity of the number of $k \in 1..n$ such that $\nu_k = \nu'_k = 1$, that is, the parity of the number of 1's in $\nu \cdot \nu'$ (component-wise binary product).

Remark. Classical computations happen in B^n. Quantum computations happen in $\mathbb{H}^{(n)}$, where the binary space B^n appears just as indices for the basis $\{\mathsf{j}^\nu\}$ of $\mathbb{H}^{(n)}$.

Remark. In Dirac notation j^ν can be denoted by $|\nu\rangle$. In this notation, $|\nu\rangle \cdot |\nu'\rangle = \pm|\nu + \nu'\rangle$.

The *Hadamard q-vector of order n* is defined as

$$\boldsymbol{h}^{(n)} = \rho^n \sum_{\nu \in B^n} \mathrm{j}^\nu,$$

where $\rho = 1/\sqrt{2}$. Since the norm squared of $\sum_{\nu \in B^n} \mathrm{j}^\nu$ is $|B^n| = 2^n$, the factor ρ^n insures that $\boldsymbol{h}^{(n)}$ is a unit vector.

We also have the expression

$$\boldsymbol{h}^{(n)} = \rho^n (\mathrm{j}^0 + \mathrm{j}^1) \otimes \overset{n)}{\cdots} \otimes (\mathrm{j}^0 + \mathrm{j}^1).$$

Indeed, to expand this product we have to choose 0 or 1 in each factor, which makes for 2^n choices, and for the choice $\nu = \nu_0, \ldots, \nu_n$ we get j^ν.

In Dirac notation:
$\boldsymbol{h}^{(n)} = \rho^n \sum_{\nu \in B^n} |\nu\rangle = \rho^n (|0\rangle + |1\rangle) \otimes \overset{n)}{\cdots} \otimes (|0\rangle + |1\rangle)$.

An element $\xi \in \mathbb{H}^{(n)}$ is said to be *split* (or *composite*) if it is of the form $\xi = \mathfrak{q}_1 \otimes \cdots \otimes \mathfrak{q}_n$, with $\mathfrak{q}_1, \ldots, \mathfrak{q}_n \in \mathbb{H}$.

The ν component of this element is $\xi_\nu = \xi_{\nu_1}(\mathfrak{q}_1) \cdots \xi_{\nu_n}(\mathfrak{q}_n)$, where we set, for $\mathfrak{q} \in \mathbb{H}$, $\mathfrak{q} = \xi_0(\mathfrak{q}) + \xi_1(\mathfrak{q})\mathrm{j}$. Now *these ξ_ν are not independent*. Indeed, we can write relations among them as follows.

Partition the ν's into those that begin with 0 and those that begin with 1. Then form the $2 \times 2^{n-1}$ matrix whose rows correspond to the ξ_ν's of these two groups. Since the two rows are proportional, all the 2×2 minors of the matrix vanish. These are the *Segre relations* and it happens that they are also sufficient (and in general redundant) to insure that a vector $\xi \in \mathbb{H}^{(n)}$ is split. N

For $n = 2$, we get a single relation: $\det \begin{bmatrix} \xi_{00} & \xi_{01} \\ \xi_{10} & \xi_{11} \end{bmatrix} = 0$. For $n = 3$ we have the matrix $\begin{pmatrix} \xi_{000} & \xi_{001} & \xi_{010} & \xi_{011} \\ \xi_{100} & \xi_{101} & \xi_{110} & \xi_{111} \end{pmatrix}$ and 6 relations.

The vectors that are not split are said to be *entangled*. For $n = 2$, the vector $\xi^{\text{EPR}} = \mathrm{j}^{00} + \mathrm{j}^{11} = |00\rangle + |11\rangle$ is entangled. A random element of $\mathbb{H}^{(n)}$, $n \geqslant 2$, is entangled, in the (technical) sense that the split vectors form a set of measure zero.

By definition, $\Sigma^{(n)} = \mathbb{H}^{(n)} - \{0\}/\sim$, a space of complex dimension $2^n - 1$. Let

$$\kappa : \mathbb{H}^{(n)} - \{0\} \to \Sigma^{(n)}$$

be the ket map, which by definition is onto and satisfies $\kappa(\xi) = \kappa(\xi')$ if and only if $\xi \sim \xi'$.

The condition for $\xi \in \mathbb{H}^{(n)}$ to be a unit vector is that

$$\sum_{\nu \in B^n} |\xi_\nu|^2 = 1.$$

This equation represents the *unit sphere* of the Euclidean space $\mathbb{H}^{(n)}_{\mathbb{R}}$. Since this Euclidean space has of dimension $2 \times 2^n = 2^{n+1}$, that sphere is denoted by $S^{2^{n+1}-1}$, and thus

$$\Sigma^{(n)} = S^{2^{n+1}-1} / \equiv.$$

The map $\kappa : S^{2^{n+1}-1} \to \Sigma^{(n)}$ is onto and with the property that $\kappa(\xi) = \kappa(\xi')$ if and only if $\xi \equiv \xi'$.

For $n = 1, 2, 3$ the (real) dimension of these spheres is $3, 7, 15$ and hence the real dimension of $\Sigma^{(n)}$ is $2, 6, 14$.

A state $\kappa(\xi)$ is said to be *split* (or *composite*) if ξ is a split vector. This is well defined, because if ξ is split and $\xi \sim \xi'$, then ξ' is split.

Let $\bar{\Sigma}^{(n)} \subset \Sigma^{(n)}$ be the set of split states. We have an onto map $(S^2)^n \to \Sigma'_n$ defined by

$$(v_1, \ldots, v_n) \mapsto \kappa(\check{v}_1 \otimes \cdots \otimes \check{v}_n).$$

This shows that entangled states are specified by $2n$ real parameters, or n complex parameters, whereas general states are specified by $2^n - 1$ complex parameters. This again confirms the assertion that *a random state is entangled*.

q-Computing

Unitary dinamics ▪ q-computations

Richard Feynman and Yuri Manin

The evolution of the system H in a time interval $[0, t]$ is governed by a *unitary operator* U_t, in the sense that if $\xi_0 \in H$ represents the state of the system at time $t = 0$, then $U_t\xi_0$ represents the state of the system at time t.

Note that $\langle U_t\xi_0|U_t\xi_0'\rangle = \langle\xi_0|\xi_0'\rangle$, in particular $|U_t\xi_0| = |\xi_0|$.

Remark. Let U be a unitary operator, A an observable and $\xi \in H$ a unit vector. Let $a_1, \dots, a_r$ be the eigenvalues of A, $E_1, \dots, E_r$ the corresponding eigenspaces, and $\xi = \xi_1' + \cdots + \xi_r'$ with $\xi_k' \in E_k$. Then (1) $A' = UAU^\dagger$ is an observable; (2) its eiganvalues are $a_1, \dots, a_r$ and its eigenspaces $UE_1, \dots, UE_r$; and (3) $U\xi = U\xi_1' + \cdots + U\xi_r'$, with $U\xi_k' \in UE_k$ and $|U\xi_k'|^2 = |\xi_k'|^2$.

If $U_t = e^{\mathrm{i}\mathfrak{h}t}$, where $\mathfrak{h}$ is an observable, we say that the evolution is *hamiltonian*, and that $\mathfrak{h}$ is the *hamiltonian* of the system. Notice that it is indeed a unitary operator: $e^{\mathrm{i}\mathfrak{h}t}(e^{\mathrm{i}\mathfrak{h}t})^\dagger = e^{\mathrm{i}\mathfrak{h}t}e^{-\mathrm{i}\mathfrak{h}^\dagger t} = I$.

If the evolution of the system is hamiltonian and the hamiltonian $\mathfrak{h}$ does not depend on t, then the state vector $x = U_t x_0$ satisfies the *Schrödinger equation*: $\dot{x} = \mathrm{i}\mathfrak{h}x$.

Thus $dx = i\mathfrak{h}\ dt\ x$. If we fix $dt = t/N$ (N large), the loop

```
x = x0
do N: x = (1 + i h dt)x
return x
```

computes an approximation of $x = U_t(x_0)$.

A q-*computation of order* n is a unitary operator $U : \mathbb{H}^{(n)} \to \mathbb{H}^{(n)}$. With the composition, these operators form the unitary group of $\mathbb{H}^{(n)}$ that here will be denoted by $\mathcal{U}^{(n)}$.

- *Identity*: $\mathrm{Id} \in \mathcal{U}^{(n)}$.
- *Composition*: If $U, V \in \mathcal{U}^{(n)}$. then $UV \in \mathcal{U}^{(n)}$.
- *Reversibility*: If $U \in \mathcal{U}^{(n)}$, then $U^{-1} = U^{\dagger} \in \mathcal{U}^{(n)}$.

Examples. (1) If $U \in \mathcal{U}^{(n)}$ and $U' \in \mathcal{U}^{(n')}$, then $U \otimes U' \in \mathcal{U}^{(n+n')}$.

(2) If $U_1, \ldots, U_n \in \mathcal{U}^{(1)}$, then $U_1 \otimes \cdots \otimes U_n \in \mathcal{U}^{(n)}$.

(3) If $U \in \mathcal{U}^{(1)}$, then $U^{\otimes n} \in \mathcal{U}^{(n)}$.

(4) A reversible classical computation on n bits is a bijective map $f : B^n \to B^n$. Associate to f the linear map $U_f = \mathbb{H}^{(n)} \to \mathbb{H}^{(n)}$ uniquely defined by $U_f(\mathrm{j}^{\nu}) = \mathrm{j}^{f(\nu)}$, or $U_f(|\nu\rangle) = |f(\nu)\rangle$. Since $\nu \mapsto f(\nu)$ is a permutation of the ν's, U_f permutes the $\mathrm{j}^{\nu} = |\nu\rangle$, so it is unitary, and hence a q-computation of order n.

The matrix of a *q*-computation U with respect to the orthonormal basis $\{\mathrm{j}^\nu = |\nu\rangle\}$ is the unitary matrix $\boldsymbol{U} = (u^\nu_{\nu'})_{\nu,\nu'\in B^n}$ defined by

$$U(\mathrm{j}^\nu) = \sum_{\nu'\in B^n} u^\nu_{\nu'}\mathrm{j}^{\nu'}, \quad \text{or} \quad U(|\nu\rangle) = \sum_{\nu'\in B^n} u^\nu_{\nu'}|\nu'\rangle.$$

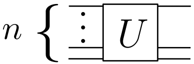
These unitary matrices form a group, $\boldsymbol{\mathcal{U}}^{(n)}$, with the multiplication operation, and the map $\mathcal{U}^{(n)} \to \boldsymbol{\mathcal{U}}^{(n)}$, $U \mapsto \boldsymbol{U}$, is an isomorphism.

If $\boldsymbol{\xi}$ is the row of components of $\xi \in \mathbb{H}^{(n)}$, and $\mathbf{j}$ the column formed with the j^ν, then we have (using Einstein's summation criterion) that

$$U(\xi) = \xi_\nu U(\mathrm{j}^\nu) = \xi_\nu u^\nu_{\nu'}\mathrm{j}^{\nu'} = \boldsymbol{\xi}\boldsymbol{U}\mathbf{j}.$$

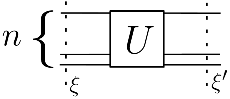
This means that the row of components of $U(\xi)$ is $\boldsymbol{\xi}\boldsymbol{U}$.

A q-computation U of order n is often represented by a diagram like this:

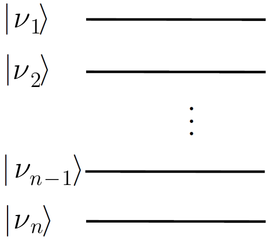


The n horizontal lines are called q-*wires*. Each wire carries a q-bit state.

If we want to represent the q-input ξ and q-output ξ', the diagram can be modified as follows:



In the case where $\xi = |\nu_1\rangle \cdots |\nu_n\rangle$, the input is represented as follows:

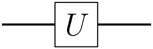


The only classical computations on one bit ν are the identity and the negation. If we denote the negation by $\textsc{Not}$, we can write $\textsc{Not}(\nu) = 1 + \nu$. Thus $\textsc{Not}(0) = 1$ and $\textsc{Not}(1) = 0$. The q-computation defined by $\textsc{Not}$ is the operator X defined by $X(\mathrm{j}^{\nu}) = \mathrm{j}^{1+\nu}$ or $X(|\nu\rangle) = |1 + \nu\rangle$ (QModels, p.51, (1)).

In contrast to the classical computations, the q-computations of order 1 are given by unitary operators of $\mathbb{H}$. In matrix form, these operators have the form

$$U = e^{\mathrm{i}\alpha} \begin{bmatrix} z & w \\ -\bar{w} & \bar{z} \end{bmatrix},$$

where $\alpha \in [0, 2\pi)$, $z, w \in \mathbb{C}$, and $z\bar{z} + w\bar{w} = 1$.



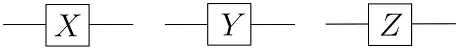
In addtion to X, we have the operators Y and Z defined in *loc. cit.* The significance of X, Y, Z in relation to rotations of E_3 is that (QModels, p.52, (3)):

$$U_{\check{u}_x,\alpha} = e^{\mathrm{i}\alpha X} \simeq \begin{bmatrix} \cos\alpha & \mathrm{i}\sin\alpha \\ \mathrm{i}\sin\alpha & \cos\alpha \end{bmatrix}$$

$$U_{\check{u}_y,\alpha} = e^{\mathrm{i}\alpha Y} \simeq \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

$$U_{\check{u}_z,\alpha} = e^{\mathrm{i}\alpha Z} \simeq \begin{bmatrix} e^{-\mathrm{i}\alpha} & \\ & e^{\mathrm{i}\alpha} \end{bmatrix}.$$

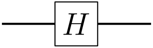
These q-computations induce the rotations of S^2 about u_x, u_y, u_z of amplitude 2α. They are hamiltonian (with respect to α) with hamiltonians X, Y, Z.



The Hadamard gate is defined as follows:

$\textsc{Had}(\mathsf{j}^\nu) = \frac{1}{\sqrt{2}}(\mathsf{j}^0 + (-1)^\nu \mathsf{j}^1)$ or
$\textsc{Had}(|0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $\textsc{Had}(|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Its matrix is $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.



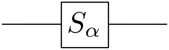
The *phase shift gate* S_α $(\alpha \in [0, 2\pi))$ is defined as follows:

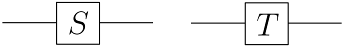
$\mathrm{S}_\alpha(\mathrm{j}^0) = \mathrm{j}^0$ and $\mathrm{S}_\alpha(\mathrm{j}^1) = e^{\mathrm{i}\alpha}\mathrm{j}^1$, or
$\mathrm{S}_\alpha(|0\rangle) = |0\rangle$ and $\mathrm{S}_\alpha(|1\rangle) = e^{i\alpha}|1\rangle$.

Its matrix is $\mathrm{diag}(1, e^{\mathrm{i}\alpha})$.

S_α can be expressed as $e^{\mathrm{i}P_{\mathrm{j}}\alpha}$, $P_{\mathrm{j}} \approx \mathrm{diag}(0, 1)$. Indeed,
$e^{\mathrm{i}\alpha\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}} = e^{\begin{bmatrix}0 & 0\\ 0 & \mathrm{i}\alpha\end{bmatrix}} = \begin{bmatrix}1 & \\ & e^{\mathrm{i}\alpha}\end{bmatrix}$. So S_α is hamiltonian in α.

Special cases: $\mathrm{S} = \mathrm{S}_{\pi/2} \approx \begin{bmatrix}1 & \\ & \mathrm{i}\end{bmatrix}$ and $\mathrm{T} = \mathrm{S}_{\pi/4} \approx \begin{bmatrix}1 & \\ & e^{\mathrm{i}\pi/4}\end{bmatrix}$.



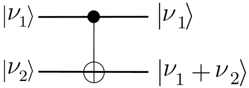


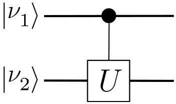
The *controlled*-NOT *gate* is the *q*-computation of order 2, CNOT, defined as follows:

$\mathrm{CNOT}(\mathsf{j}^{\nu_1}\mathsf{j}^{\nu_2}) = \mathsf{j}^{\nu_1}\mathsf{j}^{\nu_1+\nu_2}$, or

$\mathrm{CNOT}(|\nu_1\nu_2\rangle) = |\nu_1(\nu_1+\nu_2)\rangle$.

If $\nu_1 = 0$, it does nothing, and when $\nu_1 = 1$ it negates the second *q*-bit. It corresponds to the classical computation $B^2 \to B^2$, $(\nu_1,\nu_2) \mapsto (\nu_1, \nu_1+\nu_2)$.





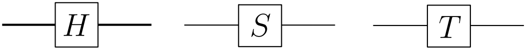
1. *U-gates*, $U \in \mathcal{U}^{(1)}$

For $k \in 1..n$, let U_k denote the action of U on the k-th q-bit. More precisely, it is the q-computation defined as follows:

$$|\cdots\nu_k\cdots\rangle = |\cdots\rangle\,|\nu_k\rangle\,|\cdots\rangle \mapsto |\cdots\rangle\; U|\nu_k\rangle\,|\cdots\rangle.$$

For $n = 2$, for instance, $U_2 = \mathsf{Id} \otimes U$ and if $n = 3$, $U_2 = \mathsf{Id} \otimes U \otimes \mathsf{Id}$.

An U-gate will be called *restricted* if U is chosen from $\{H, S, T\}$, where $H = \mathrm{H{\scriptsize AD}}$ $\mathrm{S} = \mathrm{S}_{\pi/2}$, $\mathrm{T} = \mathrm{S}_{\pi/4}$.

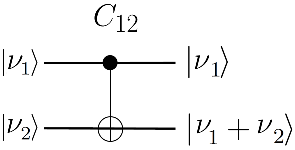


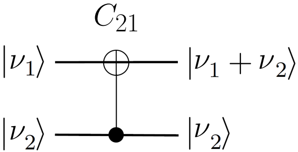
The definition of U_K when K is a list of integers in $1..n$ is straightforwar: Repeat U_k for each $k \in K$.

2. CNOT-gates, $C_{r,s}$

Given $r, s \in 1..n$, $C_{r,s}$ is the q-computation that negates the s-th q-bit if (an only if) the r-th q-bit is $|1\rangle$. In other words, it is the linear map which is the identity on the basis q-vectors of the form $|\cdots 0_r \cdots\rangle$ and such that

$$|\cdots 1_r \cdots 0_s \cdots\rangle \mapsto |\cdots 1_r \cdots 1_s \cdots\rangle$$
$$|\cdots 1_r \cdots 1_s \cdots\rangle \mapsto |\cdots 1_r \cdots 0_s \cdots\rangle$$





Example

The q-computation $C_{12}(U)$ is quite different from $I_2 \otimes U$. In fact, the matrix of the latter is

$$\begin{bmatrix} u_{00} & u_{01} & 0 & 0 \\ u_{10} & u_{11} & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}$$

Notice, for example, that

$(I_2 \otimes U)|00\rangle = |0\rangle\, U|0\rangle = u_{00}|0\rangle|0\rangle + u_{10}|0\rangle|1\rangle,$

whereas $C_{12}(U)|00\rangle = |00\rangle$.

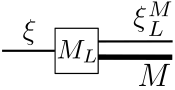
3. Measurement $M_L(\xi)$

To produce a mathematical model of a quantum computation we need to include the operation of measuring a set $L = \{l_1, \ldots, l_r\} \subseteq \{1, \ldots, n\}$ of q-bits.

Let $\xi \in \mathbb{H}^{(n)}$ be a unit q-vector representing the current state of a *q-register of length n* (the *q-memory*).

Consider the observable A_L uniquely defined by $A_L(|\nu\rangle) = \nu_L|\nu\rangle$, where $\nu_L = \nu_{l_1} \cdots \nu_{l_r}$ (viewed as an r-bit integer). Its eigenvalues are the r-bit integers M, and the M-eigenspace is the space $E_M \subseteq \mathbb{H}^{(n)}$ spanned by all $|\nu\rangle$ such that $\nu_L = M$.

The orthogonal projection of ξ on E_M is $\xi_L^M = \sum_{\nu_L=M} \xi_\nu|\nu\rangle$ (L-*collapses* of ξ) and the probability of getting M is $p_M = |\xi_L^M|^2$.



In the case when $L = \{1, \ldots, n\}$, we write simply $M(\xi)$. In that case the possible outcomes are the elements $\nu \in B^n$ and the corresponding collapses are $\xi^{\nu}_{1..n} = \xi_\nu|\nu\rangle$, with probabilities $|\xi_\nu|^2$. In this context, the coefficient ξ_ν is usually called the (probability) *amplitude* of $|\nu\rangle$, and the probability of this result is $p_\nu = |\xi_\nu|^2$: *the probability is the norm squared of the amplitude*.

If $n = 3$, for instance, then there are eight possible outcomes for $M(\xi) = M_{123}(\xi)$ and the corresponding collapses are $\xi^{rst}_{123} = \xi_{rst}|rst\rangle$ ($rst \in B^3$) with probabilities $p_{rst} = |\xi_{rst}|^2$.

A *q*-*procedure* is a sequence of actions, each of which is either a *q*-computation or a *q*-measurement, that are applied successively to $|0\cdots0\rangle$ (this is the *default* initial state of the *q*-*memory*). Since procedures are meant to produce results, usually the last action is a (full) *q*-measurement.

Example: random number generator

The following *q*-procedure outputs random numbers of n bits with a uniform probability distribution:

RANDOM

$$\xi = H^{\otimes n}|0\ldots0\rangle = H|0\rangle\cdots H|0\rangle, \quad M(\xi) \blacksquare$$

Indeed, we have seen that ξ is the Hadamard *q*-vector $\boldsymbol{h}^{(n)} = \rho\sum_\nu|\nu\rangle$ ($\rho = 1/2^{n/2}$); the amplitude of any ν is ρ; so its probability is $p_\nu = \rho^2 = 1/2^n$.

A *q-algorithm* is a q-procedure made up of only basic q-procedures.

The *complexity* of a q-algorithm is the number of basic q-procedures that compose it.

A q-algorithm is said to be

- *internal* if it does not contain q-measurements;
- *exact* if the probability of its output is 1, and *probabilistic* otherwise.
- *restricted* if the only U-gates used belong to $\{H, S, T\}$.

Theorem (Universality of the U and $\textsc{Cnot}$ gates)

1) *Any q-computation can be realized by an internal q-algorithm*.
2) *For any q-computation U there exists a restricted internal q-algorithm which approximates U to any wanted degree*.

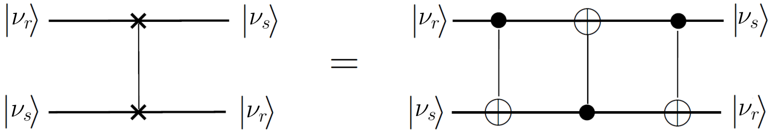
Proof See Universality (1) and Universality (2).

In the remainder of this section, we provide elementary illustrations of this theorem: $\textsc{Swap}[r,s]$, $\textsc{Had}[L]$, $\textsc{Euler}[r,s,U]$ (for $C_{r,s}(U)$), and the $\textsc{Toffoli}$ and $\textsc{Fredkin}$ gates.

Example (SWAP[r, s]) This internal q-algorithm is defined as follows:

SWAP[r, s]

$C_{r,s}$, $C_{s,r}$, $C_{r,s}$ ∎



The q-computation performed by this algorithm amounts to interchanging the states of the r-th and s-th q-bits, which means that it is equal to the linear map uniquely defined by

$$|\cdots\nu_r\cdots\nu_s\cdots\rangle \mapsto |\cdots\nu_s\cdots\nu_r\cdots\rangle$$

This statement is a direct consequence of the fact that it holds for classical computations. Indeed, for any pair of bits, (x, y), we have:

$$\begin{aligned}
&C_{1,2}(x, y) = (x, x + y),\\
&C_{2,1}(x, x + y) = (x + x + y, x + y) = (y, x + y),\\
&C_{1,2}(y, x + y) = (y, y + x + y) = (y, x).
\end{aligned}$$

Example (Multiple H) Consists in applying the Hadamard gate H at any index on a given list $L \subseteq \{1, \ldots, n\}$ of positions (denoted by H_L before):

$\textsc{Had}[L]$

for $l \in L$ do $R_l(H)$ ▪

Remark that if $m \in \{1, \ldots, n\}$, $\textsc{Had}[\{1, \ldots, m\}]$ yields a q-algorithm for the q-procedure $|\nu\rangle \mapsto (H^{\otimes m}|\nu_1 \cdots \nu_m\rangle)|j_{m+1} \cdots j_n\rangle$. This algorithm will be denoted $\textsc{Had}[m]$. In the case $m = n$, it is a q-algorithm for $H^{\otimes n}$ and instead of $\textsc{Had}[n]$ we will simple write $\textsc{Had}$.

Similar algorithms can be devised replacing H by any $U \in \boldsymbol{U}^{(1)}$. For example, $U^{\otimes n}$ can be computed by the following q-algorithm:

for $l \in \{1, \ldots, n\}$ do $R_l(U)$ ▪

We know that

$$R_z(\varphi) = \cos\tfrac{\varphi}{2} I_2 - i\sin\tfrac{\varphi}{2} Z = e^{-i\frac{\varphi}{2}Z}$$
$$R_y(\theta) = \cos\tfrac{\theta}{2} I_2 - i\sin\tfrac{\theta}{2} Y = e^{-i\frac{\theta}{2}Y}$$
$$R_x(\psi) = \cos\tfrac{\psi}{2} I_2 - i\sin\tfrac{\psi}{2} X = e^{-i\frac{\psi}{2}X}$$

Given $U \in \boldsymbol{U}^{(1)}$, it can be expressed as

$$U = e^{i\alpha} AXBXC,$$

with $A, B, C \in S\boldsymbol{U}^{(1)}$ and $ABC = I_2$ (this will be called an *Euler decomposition of* U). Indeed, there are (Euler) angles $\alpha, \beta, \theta, \gamma$ such that $U = e^{i\alpha} R_z(\beta) R_y(\theta) R_z(\gamma)$, and it is enough to set

$$A = R_z(\beta) R_y(\tfrac{\theta}{2})$$
$$B = R_y(-\tfrac{\theta}{2}) R_z(-\tfrac{\beta+\gamma}{2})$$
$$C = R_z(\tfrac{\gamma-\beta}{2})$$

The proof is a staightforward computation.

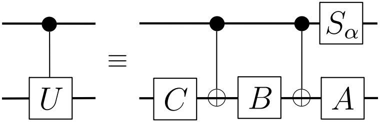
Algorithm for $C_{r,s}(U)$

$\mathrm{E{\scriptsize ULER}}[r, s, U]$

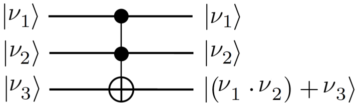
$R_s(C), C_{r,s}, R_s(B), C_{r,s}, R_s(A), R_r(S_\alpha)$ ▪

Proof. We may assume $r = 1$ and $s = 2$, as the argument can be easily adapted to the general case. Note that if $\nu_1 = 0$, then the $C_{1,2}$ and $R_1(S_\alpha)$ act as the identity and hence, since $ABC = I_2$, $\mathrm{E{\scriptsize ULER}}[1, 2, U]$ also acts as the identity. If $\nu_1 = 1$, then the action on $|\nu_2\rangle$ is $AXBXC(|\nu_2\rangle)$ and on $|\nu_1\rangle = |1\rangle$ by the phase factor $e^{i\alpha}$:

$$|1\rangle|\nu_2\rangle \mapsto e^{i\alpha}|1\rangle\, AXBXC|\nu_2\rangle = |1\rangle\, U|\nu_2\rangle.$$

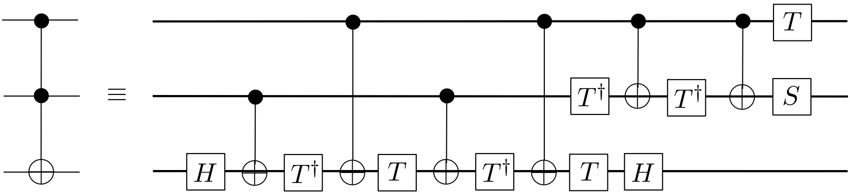


The *Toffoli gate* is the q-computation of order 3 corresponding to the classical $\textsc{Nand}$ gate $\nu_1\nu_2\nu_3 \mapsto (\nu_1 \cdot \nu_2) + \nu_3$. It negates the bit ν_3 precisely when $\nu_1 = \nu_2 = 1$, so it is a *doubly controlled negation*.



It interchanges $|110\rangle$ and $|111\rangle$, leaving all other basis vectors fixed. It follows that its matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

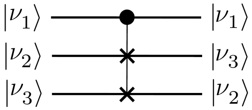


TOFFOLI

$R_3(H)$, $C_{2,3}$, $R_3(T^\dagger)$, $C_{1,3}$, $R_3(T)$, $C_{2,3}$, $R_3(T^\dagger)$, $C_{1,3}$, $R_3(T)$, $R_3(H)$, $R_2(T^\dagger)$, $C_{1,2}$, $R_2(T^\dagger)$, $C_{1,2}$, $R_2(S)$, $R_1(T)$ ∎

The *Fredkin gate* is the q-computation of order 3 corresponding to the classical computation $0\nu_2\nu_3 \mapsto 0\nu_2\nu_3$ and $1\nu_2\nu_3 \mapsto 1\nu_3\nu_2$. In other words, it is a *controlled-swap*. It interchanges $|110\rangle$ and $|101\rangle$ and leaves all other basis vectors fixed. Hence its matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



In this example we indicate how to obtain a q-algorithm for the q-procedure $C_{\{1,\ldots,n\},n+1}(U)$ defined by the relations

$$|\nu\rangle|\nu_{n+1}\rangle \mapsto \begin{cases} |\mathbf{1}_n\rangle|1+\nu_{n+1}\rangle & \text{if } j = \mathbf{1}_n \\ |\nu\rangle|\nu_{n+1}\rangle & \text{otherwise} \end{cases}$$

If we take $V \in \boldsymbol{U}^{(1)}$ such that $U = V^2$, then the algorithm is based on the following recursive recipe:

$\textsc{Control}[\{1,\ldots,n\}, n+1, U]$

$\quad\textsc{Control}[\{2,\ldots,n\}, n+1, V]$

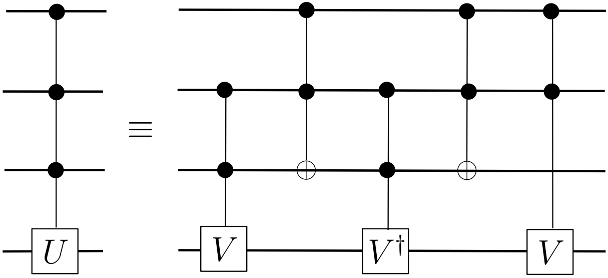
$\quad\textsc{Control}[\{1,\ldots,n-1\}, n, X]$

$\quad\textsc{Control}[\{2,\ldots,n\}, n+1, V^\dagger]$

$\quad\textsc{Control}[\{1,\ldots,n-1\}, n, X]$

$\quad\textsc{Control}[\{1,\ldots,n-1\}, n+1, V]$ ∎

In other words, an n-controlled U-gate is reduced to five $(n-1)$-controlled U-gates. This is more easily grasped pictorially. Consider, for instance, the case $n = 3$:



If the first q-bit is $|0\rangle$, then the action on the fourth q-bit is $VV^\dagger = I_2$. If the second q-bit is $|0\rangle$, then the action on the fourth q-bit is I_2. If the third q-bit is $|0\rangle$, and the first and second are $|1\rangle$, then the action on the fourth q-bit is $V^\dagger V = I_2$. Finally, if the three q-bits are $|1\rangle$, then the action on the fourth q-bit is $V^2 = U$.

Consider the plane $P = [|\nu\rangle, |\nu'\rangle]$ spanned by two different basis vectors $|\nu\rangle$ and $|\nu'\rangle$. Then we can let $U = [[a, b], [c, d]] \in S\boldsymbol{U}^{(1)}$ act on that plane in the obvious way: $U|\nu\rangle = a|\nu\rangle + b|\nu'\rangle$ and $U|\nu'\rangle = c|\nu\rangle + d|\nu'\rangle$. Moreover, we can extend this action to $\mathbb{H}^{(n)}$ so that $U|\nu''\rangle = |\nu''\rangle$ for all $\nu'' \neq \nu, \nu'$. Since $|\nu''\rangle$ is orthogonal to P, this action is a q-computation (we will write $U_{\nu,\nu'}$ to denote it). Note, for example, that if $\nu = 1$ and $\nu' = 2$, then the matrix of our q-computation is $U \oplus I_{2^n-2}$.

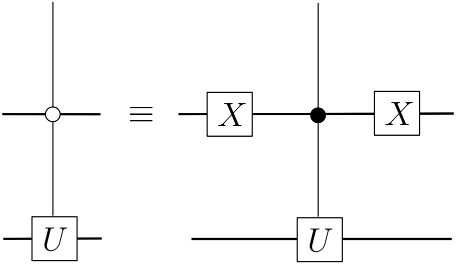
Let us indicate how to get a q-algorithm for $U_{\nu,\nu'}$. In fact, by the previous example, it will be enough to show how to resolve $U_{\nu,\nu'}$ by means of simple and multicontrol U-gates.

The simplest case is when $|\nu\rangle$ and $|\nu'\rangle$ have the form

$$|\nu\rangle = |x\rangle|0\rangle|y\rangle, \quad |\nu'\rangle = |x\rangle|1\rangle|y\rangle.$$

Indeed, in this case
$U|\nu\rangle = a|\nu\rangle + b|\nu'\rangle = |x\rangle(a|0\rangle + b|1\rangle)|y\rangle = |x\rangle(U|0\rangle)|y\rangle$,
$U|\nu'\rangle = |x\rangle(U|1\rangle)|y\rangle$ (similar computation), and therefore $U_{\nu,\nu'}$ is a multicontrol U-gate, in the sense that $U_{\nu,\nu'}(|x'\rangle|t\rangle|y'\rangle) = |x'\rangle|t\rangle|y'\rangle$ if $x \neq x'$ or $y \neq y'$ and otherwise it is equal to $|x\rangle(U|t\rangle)|y\rangle$.

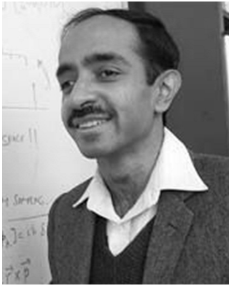
Note that if the controlling value of a bit is 0, then we can reduce it to the standard controlling value 1 and two X gates, as shown in the picture (the white circle is to indicate that the control value is 0):



If ν and ν' differ in $r \geqslant 2$ places, let $\nu'' \in B^n$ be such that ν'' differs in one position from ν and in $r-1$ positions from ν'. By induction we may assume that there is a q-algorithm to compute $U_{\nu'',\nu'}$, for the case $r=1$ has already been established. Now a q-algorithm for $U_{\nu,\nu'}$ is obtained on noticing that it coincides with $X_{\nu,\nu''}U_{\nu'',\nu'}X_{\nu,\nu''}$, where $X_{\nu,\nu''}$ is defined so that $X_{\nu,\nu''}|\nu\rangle = |\nu''\rangle$, $X_{\nu,\nu''}|\nu''\rangle = |\nu\rangle$ and $X_{\nu,\nu''}|\lambda\rangle = |\lambda\rangle$ if $\lambda \neq \nu, \nu''$. Since $X_{\nu,\nu''}$ is a (form of) multicontrol-$\textsc{Not}$, it can be computed by a q-algorithm, and thus so it can $U_{\nu,\nu'}$.

Two archtypal q-algorithms

Deutsch-Josza's q-algorithm
Grover's searching q-algorithm



Lov Kumar Grover

Let $f : B^n \to B$ be a map, and assume we know that it is either *constant* or *balanced* (this means that the sets $f^{-1}(0)$ and $f^{-1}(1)$ have the same cardinal). Then the problem consists in deciding which of the two possibilities holds.

Remark

The classical solution is based on evaluating f on successive elements of B^n. This process stops as soon as either we have found two different values, in which case f has to be balanced, or else the number of evaluations has exceeded 2^{n-1}, in which case f must be constant. Since the worse case requires $2^{n-1}+1$ evaluations, the complexity of this procedure is exponential in n.

1. Initialize a q-computer of order $n+1$ with $\sigma_1 = |0\rangle \overset{n)}{\cdots} |0\rangle|1\rangle$.

2. Let $\sigma_2 = H^{\otimes(n+1)}\sigma_1 = \rho^{n+1}\sum_{\nu=0}^{2^n-1} |\nu\rangle(|0\rangle - |1\rangle)$.

3. Let $U_{\tilde{f}}$ be the q-computation corresponding to the classical (reversible) computation $\tilde{f} : B^n \times B \to B^n \times B$, $(\nu, b) \mapsto (\nu, b + f(\nu))$ and let $\sigma_3 = U_{\tilde{f}}\sigma_2$. Since

$$U_{\tilde{f}}(|\nu\rangle|b\rangle) = |\nu\rangle|b + f(\nu)\rangle$$

we clearly have

$$\sigma_3 = \rho^{n+1}\sum_{\nu=0}^{2^n-1} |\nu\rangle(|f(\nu)\rangle - |1 + f(\nu)\rangle).$$

Note that this can be written as

$$\rho^{n+1}\sum_{\nu\in B^n}^{2^n-1}(-1)^{f(\nu)}|\nu\rangle(|0\rangle - |1\rangle) = \rho^{n+1}\sum_{\nu_1,\ldots,\nu_n\in B}(-1)^{f(\nu_1\cdots\nu_n)}|\nu_1\cdots\nu_n\rangle(|0\rangle - |1\rangle).$$

4. Compute $\sigma_4 = (H^{\otimes n} \otimes I_2)\,\sigma_3$. Since

$$\begin{aligned}(H^{\otimes n} \otimes I_2)|\nu_1 \cdots \nu_n\rangle(|0\rangle - |1\rangle) &= (H|\nu_1\rangle) \cdots (H|\nu_n\rangle)(|0\rangle - |1\rangle)\\ &= \rho^n \prod_{r=1}^{n}(|0\rangle + (-1)^{\nu_r}|1\rangle)(|0\rangle - |1\rangle)\\ &= \rho^n \sum_{\nu' \in B^n} (-1)^{\nu\cdot\nu'}|\nu'\rangle(|0\rangle - |1\rangle),\end{aligned}$$

where $\nu \cdot \nu' = \nu_1\nu'_1 + \cdots + \nu_n\nu'_n$ is the scalar product of the binary vectors ν and ν', we find

$$\begin{aligned}\sigma_4 &= \rho^{2n+1} \sum_{\nu \in B^n} \sum_{\nu' \in B^n} (-1)^{\nu\cdot\nu' + f(\nu)}|\nu'\rangle(|0\rangle - |1\rangle)\\ &= \rho^{2n+1} \sum_{\nu,\nu' \in B^n} (-1)^{\nu\cdot\nu' + f(\nu)}|\nu'\rangle(|0\rangle - |1\rangle).\end{aligned}$$

Let us look at the coefficient $a_{\nu'} = \rho^{2n+1} \sum_\nu (-1)^{\nu\cdot\nu' + f(\nu)}$ of $|\nu'\rangle(|0\rangle - |1\rangle)$ in this expression.

If f is constant, $a_{\nu'} = \rho^{2n+1}(-1)^{f(0_n)} \sum_\nu (-1)^{\nu\cdot\nu'}$, so that $a_{0_n} = (-1)^{f(0_n)}\rho$ and $a_{\nu'} = 0$ for $\nu' \neq 0_n$.

If f is balanced then $a_0 = \rho^{2n+1} \sum_\nu (-1)^{f(\nu)} = 0$, and clearly $a_{\nu'} \neq 0$ for some $\nu' \neq 0_n$.

We can summarize these findings as follows:

$$\sigma_4 = \begin{cases} \rho|0_n\rangle\,(|0\rangle - |1\rangle) & \text{if} f \text{ is constant,} \\ \sum_{\nu'\neq 0} a_{\nu'}|\nu'\rangle\,(|0\rangle - |1\rangle) & \text{if} f \text{ is balanced,} \end{cases}$$

5. The last step consists in measuring the first n q-bits. If f is constant, the result is 0 with certainty, and if f is balanced, then we obtain a non-zero integer. Hence, the q-procedure decides *exactly* whether f is constant or not.

DEUTSCH[f]

$$
\begin{array}{ll}
 & \to |0_n\rangle|0\rangle \\
R_{n+1}(X) & \to |0\cdots 0\rangle|1\rangle \\
\text{HADAMARD} & \to \rho^{n+1}\sum_{\nu\in B^n}|\nu\rangle\big(|0\rangle-|1\rangle\big) \\
U_{\tilde{f}} & \to \rho^{n+1}\sum_{\nu\in B^n}(-1)^{f(\nu)}|\nu\rangle\big(|0\rangle-|1\rangle\big) \\
\text{HADAMARD}[n] & \to \rho^{2n+1}\sum_{\nu,\nu'\in B^n}(-1)^{\nu\cdot\nu'+f(\nu)}|\nu'\rangle\big(|0\rangle-|1\rangle\big)
\end{array}
$$

$//\rho|0_n\rangle(|0\rangle-|1\rangle)$ if f is constant, and

$//\sum_{\nu\neq 0_n} a_\nu|\nu\rangle(|0\rangle-|1\rangle)$ if f is balanced.

$M_{\{1,\ldots,n\}} \quad \to M$

if $M=0$ then Constant else Balanced ∎

Deutsch-Josza

GroverG GroverK

Grover search

Suppose $\{\nu \to x_\nu \,|\, \nu \in B^n\}$. If we are to search for the ν such that x_ν satisfies some condition, like finding the position of a given number in a random list, in the worst case we will have to examine all the $N = 2^n$ items. In any case, to find a randomly chosen value x we will need, on the average, $N/2$ tests.

The remarkable discovery of Grover [2, 3] is that there is a q-algorithm with complexity $O(\sqrt{N/M})$ that finds an x satisfying the condition, where M is the number of solutions to the query.

Remark. As most q-algorithms, Grover's q-algorithm is probabilistic, in the sense that there is a small probability p that the outcome of a run does not satisfy the condition. As it is customary is such cases, running the algorithm some fixed number of times k (this does not change the complexity) will yield an answer that may be wrong with probability p^k, a value that usually is negligibly small already for small k.

Let J_1 (J_0) be the subset of B^n formed with the ν such that x_ν satisfies (does not satisfy) the condition in question. Consider the map $f : B^n \to B$ such that

$$f(\nu) = \begin{cases} 0 & \text{if } \nu \in J_0 \\ 1 & \text{if } \nu \in J_1 \end{cases}.$$

Define the unit q-vectors

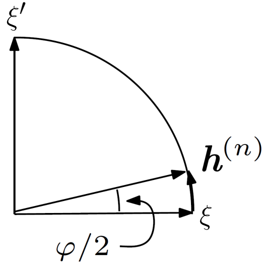
$$\xi = \tfrac{1}{\sqrt{N-M}} \sum_{\nu \in J_0} |\nu\rangle \quad \text{and} \quad \xi' = \tfrac{1}{\sqrt{M}} \sum_{\nu \in J_1} |\nu\rangle.$$

The non-zero summands in ξ' (respectively ξ) are the basis vectors corresponding to the solutions (non-solutions) of our query.

Note also that

$$\boldsymbol{h}^{(n)} = \sqrt{\tfrac{N-M}{N}}\xi + \sqrt{\tfrac{M}{N}}\xi' = \cos(\tfrac{\varphi}{2})\xi + \sin(\tfrac{\varphi}{2})\xi',$$

where the last equality defines $\varphi \in (0,\pi)$ uniquely (*Grover angle*): $\varphi = 2\arcsin(\sqrt{M/N})$.



Remark. The trigonometric formulae for the double angle imply

$$\sin(\varphi) = \tfrac{2\sqrt{M}\sqrt{N-M}}{N}, \quad \cos(\varphi) = \tfrac{N-2M}{N}.$$

To explain how Grover's procedure works, we need to introduce two auxiliary q-computations of order n, which we denote G_f and K.

The definition of G_f is as follows ($\nu \in B^n$):

$$G_f(|\nu\rangle) = \begin{cases} -|\nu\rangle & \text{if } \nu \in J_1 \\ |\nu\rangle & \text{if } \nu \in J_0 \end{cases}$$

In other words, G_f is the reflexion with respect to the space spanned by the non-solutions. In particular, $G_f(\xi) = \xi$ and $G_f(\xi') = -\xi'$. Therefore we also have

$$G_f(\boldsymbol{h}^{(n)}) = G_f\left(\cos\left(\tfrac{\varphi}{2}\right)\xi + \sin\left(\tfrac{\varphi}{2}\right)\xi'\right) = \cos\left(\tfrac{\varphi}{2}\right)\xi - \sin\left(\tfrac{\varphi}{2}\right)\xi'.$$

The q-computation K, which does not depend on f, is defined as

$$K(x) = \sum_{\nu}(2\mu_x - x_\nu)|\nu\rangle,$$

where $\mu_x = \frac{1}{N}\sum_\nu x_\nu$, the average of the amplitudes x_ν of x (we say that K is the *inversion with respect to the mean*). This linear map is indeed a q-computation, for it preserves the norm:

$$\begin{aligned}
|K(x)|^2 &= \sum_{\nu}(2\mu_x - x_\nu)(2\bar{\mu}_x - \bar{x}_\nu) \\
&= 4N\mu_x\bar{\mu}_x - 2\bar{\mu}_x\sum_{\nu} x_\nu - 2\mu_x\sum_{\nu}\bar{x}_\nu + \sum_{\nu} x_\nu\bar{x}_\nu \\
&= 4N\mu_x\bar{\mu}_x - 2N\bar{\mu}_x\mu_x - 2N\mu_x\bar{\mu}_x + |x|^2 \\
&= |x|^2.
\end{aligned}$$

Now Grover's q-procedure can be described as follows:

1. Let $u_0 = \boldsymbol{h}^{(n)} = \cos\left(\frac{\varphi}{2}\right)\xi + \sin\left(\frac{\varphi}{2}\right)\xi'$.
2. For $j = 1, \ldots, m = \lfloor \frac{\pi}{2\varphi} \rfloor$, define $u_j = K(G_f(u_{j-1}))$.
3. Return $M(u_m)$.

The main reason why this procedure works is that *in the plane spanned by ξ and ξ' the map KG_f is a rotation of amplitude φ*. Actually it is enough to show that

$$K\xi = \cos(\varphi)\xi + \sin(\varphi)\xi'$$

and

$$K(-\xi') = -\sin(\varphi)\xi + \cos(\varphi)\xi',$$

and these follow from straightforward computations using the definition of K and the formulae in Remark on page 55. Details. In particular we have

$$u_j = \xi\cos\frac{2j+1}{2}\varphi + \xi'\sin\frac{2j+1}{2}\varphi.$$

This tells us that the optimal choice for the number m of iterations in step 2 is the least positive integer such that u_m is closest to ξ', and this clearly occurs when m is the nearest integer to

$$(\frac{\pi}{2} - \frac{\varphi}{2})/\varphi = \frac{\pi}{2\varphi} - 1/2,$$

that is, when $m = \lfloor \frac{\pi}{2\varphi} \rfloor = \left\lfloor \frac{\pi}{4\arcsin(\sqrt{M/N})} \right\rfloor$.[2]

[2] We use that the nearest integer to $x - \frac{1}{2}$ is $\lfloor x \rfloor$.

Remark. Since $\arcsin(x) > x$ for $x \in (0, \frac{\pi}{2})$, we have

$$m \leqslant \frac{\pi}{4\arcsin\sqrt{M/N}} \leqslant \frac{\pi}{4}\sqrt{N/M}.$$

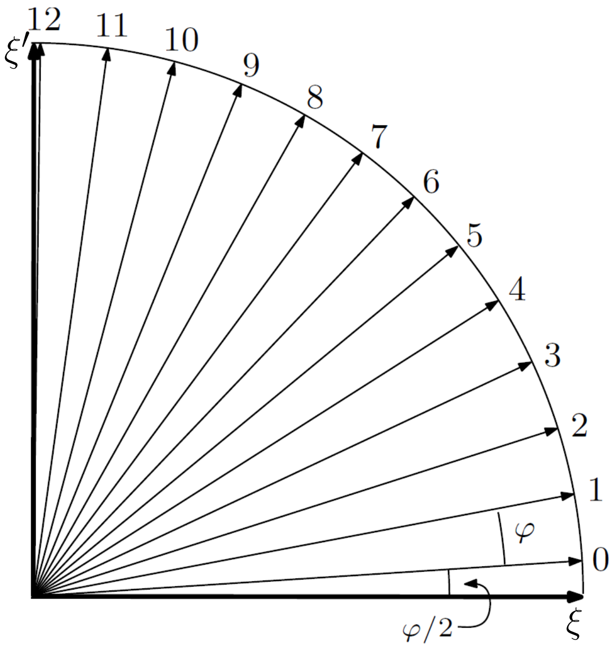
Hence also $m \leqslant \lfloor \frac{\pi}{4}\sqrt{N/M} \rfloor$. Since $\frac{\pi}{4x} - \frac{\pi}{4\arcsin(x)} < 1$ for all $x \in (0,1)$, we also have $\lfloor \frac{\pi}{4}\sqrt{N/M} \rfloor \leqslant m+1$. A more detailed study shows that when $x \to 0$ the intervals in which $\lfloor \frac{\pi}{4x} \rfloor = \lfloor \frac{\pi}{4\arcsin(x)} \rfloor + 1$ become negligibly small compared to the intervals in which $\lfloor \frac{\pi}{4x} \rfloor = \lfloor \frac{\pi}{4\arcsin(x)} \rfloor$. Thus if we iterate $\lfloor \frac{\pi}{4}\sqrt{N/M} \rfloor$ times the loop in step 2 of Grover's q-procedure, we would get the right number of rotations most of the time and otherwise we would go one step beyond, which in practice gives a q-vector that is almost as good as the previous one.

The probability of obtaining a right answer in a run of Grover's q-procedure is $p = \sin^2(\frac{2m+1}{2}\varphi)$, as $\sin(\frac{2m+1}{2}\varphi)\frac{1}{\sqrt{M}}$ is the amplitude in u_m of any of the M solutions. Similarly, the probability of obtaining an erroneous answer is $q = \cos^2(\frac{2m+1}{2}\varphi)$. Since the specification on m entails that $\frac{2m+1}{2}\varphi = \frac{\pi}{2} + \varepsilon$, with $|\varepsilon| \leqslant \varphi/2$, we see that

$$\begin{aligned} p &= \sin^2(\tfrac{\pi}{2} + \varepsilon) = \cos^2(\varepsilon) = \cos^2(|\varepsilon|) \\ &\geqslant \cos^2\left(\tfrac{\varphi}{2}\right) = \cos^2\left(\arcsin\left(\sqrt{M/N}\right)\right) = 1 - \tfrac{M}{N}. \end{aligned}$$

Hence also $q = 1 - p \leqslant M/N$.

Example. Let us illustrate the ideas so far with the case $n = 8$ and $M = 1$. We get $N = 256$, $\varphi = 7.166643^\circ$, $m = 12$. The slope of the vector u_{12} is 89.583042° and the probability of success is $p = 0.999947$. Note that p is much closer to 1 than the lower bound $1 - M/N = 1 - 1/256 = 0.996094$. The probability of error is $q = 0.000053$, again much closer to 0 than the upper bound $M/N = 1/256 = 0.003906$.



12
11
10
9
8
7
6
5
4
3
2
1
0
ξ'
ξ
φ
$\varphi/2$

Given a map $f : B^n \to B$ for which we know that $M = |f^{-1}(1)| > 0$, this q-algorithm computes Grover's q-procedure for f. We will work at order $n+1$ and we will let $\tilde{f}$ denote the classical reversible computation defined by $(x, b) \mapsto (x, b + f(x))$, $x \in B^n$, $b \in B$. As before, the corresponding q-computation will be denoted $U_{\tilde{f}}$. We will use the notations m and u_j ($j = 0, 1, \ldots, m$) from the forgoing discussion.

It is easy to phrase the sought q-algorithm $\textsc{Grover}[f]$ in terms of q-algorithms $\textsc{GroverG}[f]$ and $\textsc{GroverK}$ for G_f and K.

GROVER$[f, m]$

	$\to \lvert 0_n \rangle$
HADAMARD	$\to u_0 = \boldsymbol{h}^{(n)}$
for $j \in \{1, \ldots, m\}$ do	
GROVERK GROVERG$[f] \lvert u_{j-1} \rangle$	$\to \lvert u_j \rangle$
$M(u_m)$	$\to M$ ■

GroverG[f]

$$
\begin{array}{ll}
 & \to |x\rangle|1\rangle \\
 & //\ \text{Set } x = x^0 + x^1,\ x^i = \sum_{\nu\in J_i} x_\nu|\nu\rangle,\ \ i = 0, 1. \\
R_{n+1}(H) & \to \rho\big(|x^0\rangle|0\rangle + |x^1\rangle|0\rangle - |x^0\rangle|1\rangle - |x^1\rangle|1\rangle\big) \\
U_{\tilde f} & \to \rho\big(|x^0\rangle|0\rangle + |x^1\rangle|1\rangle - |x^0\rangle|1\rangle - |x^1\rangle|0\rangle\big) \\
 & = (|x^0\rangle - |x^1\rangle)(H|1\rangle) \\
R_{n+1}(H) & \to |G_f x\rangle|1\rangle \ \blacksquare
\end{array}
$$

GROVERK

$\rightarrow |x\rangle$

HADAMARD

for $l \in \{1, \ldots, n\}$ do

$R_l(X)$

//This loop acts as $X^{\otimes n}$

$C_{\{2,\ldots,n\},1}(Z)$

//Z to first q-bit controlled by all the others.

for $l \in \{1, \ldots, n\}$ do

$R_l(X)$

//$X^{\otimes n}$

HADAMARD $\rightarrow |K(x)\rangle$ ∎

Observations

q-Fourier transform

Don Coppersmith

$$\begin{tikzcd}
\textsc{QFT} \arrow[dr] & & \textsc{Kitaev} \arrow[dl] \\
& \textsc{Shor-Order} \arrow[d] & \\
& \textsc{Shor-Factor} &
\end{tikzcd}$$

The *Fourier transform* (FT) on $\mathbb{H}^{(n)}$ is the linear operator

$$F : \mathbb{H}^{(n)} \to \mathbb{H}^{(n)}, |\nu\rangle \mapsto f_\nu = \rho^n \sum_\lambda \xi^{\nu\lambda} |\lambda\rangle,$$

where $\xi = \xi_n = e^{i\frac{2\pi}{2^n}} = e^{i\frac{\pi}{2^{n-1}}}$.

Observe that $F \in \boldsymbol{U}^{(n)}$:

$$\langle f_\nu | f_{\nu'} \rangle = \frac{1}{2^n} \sum_\lambda \xi^{(\nu'-\nu)\lambda} = \delta_{\nu\nu'} \, ,$$

for, if $l \neq 0$,

$$\sum_{k=0}^{2^n-1} \xi^{lk} = \frac{(\xi^l)^{2^n} - 1}{(\xi^l - 1)} = 0.$$

Let us give an idea about how to produce a internal q-algorithm to obtain F.

We have, with $\rho = 1/\sqrt{2}$,

$$
\begin{aligned}
F|\nu\rangle &= \rho^n \sum_{\nu'=0}^{2^n-1} e^{\frac{2\pi i \nu \nu'}{2^n}} |\nu'\rangle \\
&= \rho^n \sum_{\nu'_1,\ldots,\nu'_n \in B} e^{2\pi i \nu \left(\frac{\nu'_1}{2^1} + \frac{\nu'_2}{2^2} + \cdots + \frac{\nu'_n}{2^n}\right)} |\nu'_1 \cdots \nu'_n\rangle \\
&= \rho^n \sum_{\nu'_1,\ldots,\nu'_n \in B} \bigotimes_{l=1}^{n} e^{\frac{2\pi i \nu \nu'_l}{2^l}} |\nu'_l\rangle \\
&= \rho^n \bigotimes_{l=1}^{n} \left(|0\rangle + e^{\frac{2\pi i \nu}{2^l}} |1\rangle\right).
\end{aligned}
$$

But

$$
\frac{\nu}{2^l} = \frac{\nu_n}{2^l} + \frac{\nu_{n-1}}{2^{l-1}} + \cdots + \frac{\nu_{n-(l-1)}}{2} + \left(\nu_l + \cdots + \nu_1 2^{n-l-1}\right).
$$

Since the part enclosed in parenthesis is an integer, the l-th tensor factor in the previous expression is equal to

$$|0\rangle + e^{i\pi \frac{\nu_n}{2^{l-1}}} \cdots e^{i\pi\nu_{n-(l-1)}}|1\rangle$$

Therefore

$$F|\nu\rangle = \rho^n \left(|0\rangle + e^{i\pi\nu_n}|1\rangle\right) \left(|0\rangle + e^{i\pi\frac{\nu_n}{2}} e^{i\pi\nu_{n-1}}|1\rangle\right) \cdots \left(|0\rangle + e^{i\pi\frac{\nu_n}{2^{n-1}}} \cdots e^{i\pi\frac{\nu_2}{2}} e^{i\pi\nu_1}|1\rangle\right). \qquad (*)$$

If we write this tensor product in reverse order, with one ρ for each factor,

$$\rho\left(|0\rangle + e^{i\pi\frac{\nu_n}{2^{n-1}}} \cdots e^{i\pi\frac{\nu_2}{2}} e^{i\pi\nu_1}|1\rangle\right) \rho\left(|0\rangle + e^{i\pi\frac{\nu_n}{2^{n-2}}} \cdots e^{i\pi\nu_2}|1\rangle\right) \cdots \cdots \rho\left(|0\rangle + e^{i\pi\nu_n/2} e^{i\pi\nu_{n-1}}|1\rangle\right) \rho\left(|0\rangle + e^{i\pi\nu_n}|1\rangle\right),$$

then for the l-th factor we have:

$$\rho\left(|0\rangle + e^{i\pi\frac{\nu_n}{2^{n-l}}}\cdots e^{i\pi\frac{\nu_{l+1}}{2}}e^{i\pi\nu_l}|1\rangle\right) = R_{n-l}\cdots R_1 H|\nu_l\rangle$$

where R_s means, for the l-th bit, $C_{l+s,l}(S_{i\pi/2^s})$. So we have the following q-algorithm:

QFT

for $l \in \{1,\dots,n\}$ do

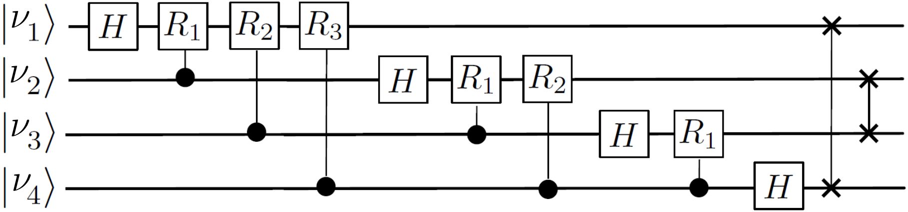
$R_l(H)$

for $s \in \{1,\dots,n-l\}$ do $C_{l+s,l}(S_{i\pi/2^s})$

for $l \in \{1,\dots,\lfloor n/2\rfloor\}$ do $\textsc{Swap}[l, n-l+1]$ ∎

This shows that QFT computes F with complexity $O(n^2)$. The swaps at the end are meant to restore the original order.

Here is a diagram to illustrate the case $n = 4$.



Remark. Let us point out, for later reference, that the formula $(*)$ can be written in the form

$$F|\nu\rangle = \rho^n(|0\rangle + e^{2\pi i\, 0.\nu_n}|1\rangle)(|0\rangle + e^{2\pi i\, 0.\nu_{n-1}\nu_n}|1\rangle)\cdots(|0\rangle + e^{2\pi i\, 0.\nu_1\cdots\nu_n}|1\rangle),$$

where, for binary digits $b_1, b_2, \ldots,$

$$0.b_1 b_2 \cdots = \frac{b_1}{2} + \frac{b_2}{2^2} + \cdots$$

Kitaev's q-phase estimation

Alexei Kitaev

Let U be a q-computation of order n, and let $u \in \mathbb{H}^{(n)}$ an eigenvector of U. The corresponding eigenvalue can be written in the form $e^{2\pi i\varphi}$, with $\varphi \in [0, 1)$. Assuming that U and u are known, then the *phase estimation problem* consists in obtaining r bits $\varphi_1, \ldots, \varphi_r$, for a given r, of the binary expansion $0.\varphi_1\varphi_2\cdots$ of φ.

The aim of this section is to phrase and analyze the interesting q-algorithm discovered by Kitaev [4] to solve this problem.

Since we need some ancillary q-bits, say m, we will work in $\mathbb{H}^{(m)} \times \mathbb{H}^{(n)}$. The algorithm assumes that we can initialize $\mathbb{H}^{(n)}$ with the q-vector u and also that we are able to perform the 'controlled' q-computations $C_{m-l+1}(U^{2^{l-1}})$, for $l = 1, \ldots, m$, defined on $\mathbb{H}^{(m)} \times \mathbb{H}^{(n)}$ as follows:

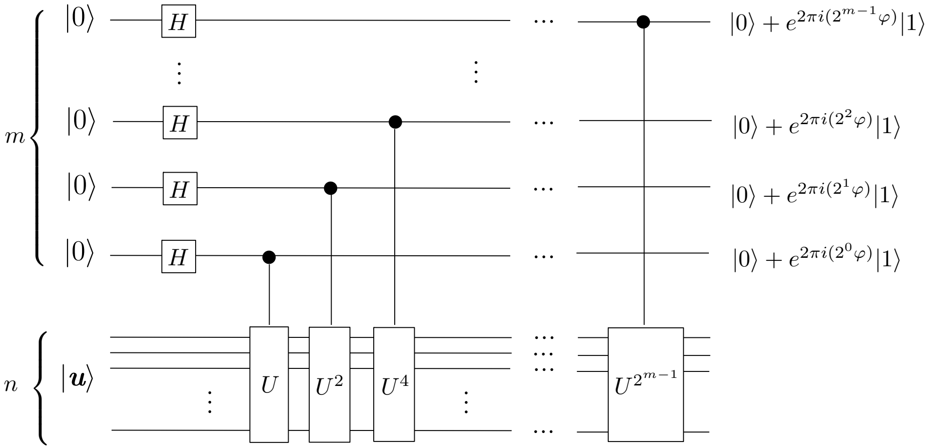
$$C_{m-l+1}(U^{2^{l-1}})\big(|\varphi_1\cdots\varphi_m\rangle|u\rangle\big) = \begin{cases} |\varphi_1\cdots\varphi_m\rangle|u\rangle & \text{if } \varphi_{m-l+1}=0 \\ |\varphi_1\cdots\varphi_m\rangle\left(U^{2^{l-1}}|u\rangle\right) & \text{if } \varphi_{m-l+1}=1 \end{cases}$$

$\textsc{Kitaev}[U, u]$

0. $\rightarrow |0_m\rangle|u\rangle$
1. $\textsc{Hadamard}[m]$ $\rightarrow |\boldsymbol{h}^{(m)}\rangle|u\rangle$
2. for $l \in 1..m$ do
   $C_{m-l+1}(U^{2^{l-1}})$
3. $\mathrm{QFT}^\dagger[m]$
4. $M_{\{1,\ldots,m\}}$

We will analyze this algorithm in two steps, denoted A and B below. In the first we will assume that $\varphi = 0.\varphi_1 \cdots \varphi_m$ and in the second we will look at the general case.

The following diagram illustrates the steps 0-2.



The action of $U^{2^{l-1}}$ only changes $|u\rangle$ by a factor, either 1 or $e^{2\pi i 2^{l-1}\varphi}$ depending on whether the controlling bit is $|0\rangle$ or $|1\rangle$. Now this factor may be moved next to the controlling bit and therefore the state at the end of the loop 2 can be written in the form

$$\rho^m \left(|0\rangle + e^{2\pi i 2^{m-1}\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{m-2}\varphi}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle\right). \tag{4}$$

This, in the notation of binary expansions, takes the form

$$\rho^m \left(|0\rangle + e^{2\pi i\, 0.\varphi_m}|1\rangle\right)\left(|0\rangle + e^{2\pi i\, 0.\varphi_{m-1}\varphi_m}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i\, 0.\varphi_1\cdots\varphi_m}|1\rangle\right), \tag{5}$$

as $e^{2\pi ik} = 1$ for any integer k. But by the Remark on page 74, this expression is equal to $F|\varphi\rangle$, which is computed by the QFT algorithm. Thus it is clear that we recover the state $|\varphi\rangle|u\rangle$ by applying $F^\dagger \otimes I_{2^n}$, where $F^\dagger$ is the inverse of F. We have denoted $\mathrm{QFT}^\dagger[m]$ the q-algorithm for $F^\dagger$ that is obtained by carrying out QFT in reverse order. Thus KITAEV supplies φ exactly in the case where φ can be expressed with m bits.

The reasoning is somewhat more involved when φ cannot be expressed using m bits. In this case, $F^\dagger \otimes I_{2^n}$ does not give the q-vector $|\varphi\rangle|u\rangle$, but a superposition of the form $\sum a_l|l\rangle|u\rangle$. As we will show below, this difficulty can be overcome in order to obtain the first r bits of φ provided $r \leqslant m$.

By expanding the product in the formula (5), we see that it can be writen in the form

$$\rho^m \sum_{k=0}^{2^m-1} e^{2\pi i\varphi k}|k\rangle|u\rangle.$$

The result in step 3 is

$$\begin{aligned}\rho^m \sum_{k=0}^{2^m-1} e^{2\pi i\varphi k}\big(F^\dagger|k\rangle\big)|u\rangle &= \rho^{2m} \sum_{k=0}^{2^m-1} e^{2\pi i\varphi k} \sum_{l=0}^{2^m-1} e^{-\frac{2\pi ikl}{2^m}}|l\rangle|u\rangle \\ &= \rho^{2m} \sum_{l=0}^{2^m-1} \left(\sum_{k=0}^{2^m-1} e^{2\pi i(\varphi-l/2^m)k}\right)|l\rangle|u\rangle \\ &= \rho^{2m} \sum_{l=0}^{2^m-1} \frac{1-e^{2\pi i(\varphi-l/2^m)2^m}}{1-e^{2\pi i(\varphi-l/2^m)}}|l\rangle|u\rangle\end{aligned}$$

Finally the result of step 4, the measurement of the first m bits, is also clear: it will be an m-bit integer l drawn with probability[3]

$$p_l = \rho^{4m}\left|\frac{1-e^{2\pi i(\varphi-l/2^m)2^m}}{1-e^{2\pi i(\varphi-l/2^m)}}\right|^2 = \rho^{4m}\frac{\sin^2 \pi\big(\varphi-l/2^m\big)2^m}{\sin^2 \pi\big(\varphi-l/2^m\big)} \qquad (*)$$

[3] We use the formula $|1-e^{i\alpha}|^2 = 4\sin^2(\alpha/2)$, which is a consequence of $|1-e^{i\alpha}|^2 = (1-e^{i\alpha})(1-e^{-i\alpha}) = 2-(e^{i\alpha}+e^{-i\alpha}) = 2(1-\cos\alpha)$.

With this distribution law we can now estimate what are the chances that the first r bits of l $(0 < r \leqslant m)$ agree with $f = \varphi_1 \cdots \varphi_r$. Indeed, using the probabilities p_l one can show (Details) that

$$p\left(|2^m\varphi - l| > 2^{m-r}\right) \leqslant \frac{1}{2(2^{m-r}-2)}. \qquad (**)$$

Therefore we can guarantee that r bits are correct with probability $1-\varepsilon$ if $\frac{1}{2(2^{m-r}-2)} \leqslant \varepsilon$, a relation that is equivalent to

$$m \geqslant r + \log_2\left(2 + \frac{1}{2\varepsilon}\right).$$

Modular order of an integer

The object of this section is a presentation of Shor's q-algorithm for finding $\mathrm{ord}_N(a)$, the order of a positive integer a modulo a positive integer N, provided $(a, N) = 1$.

By definition, $r = \mathrm{ord}_N(a)$ is the *least positive integer such that* $a^r \equiv 1 \bmod N$ or, in other words, the order of a seen as an element of the group $\mathbb{Z}_N^*$.

From a classical point of view, finding $\mathrm{ord}_N(a)$ is related to the search of the divisors of $\phi(N)$ (where ϕ denotes the classical Euler's totient function), which has exponential complexity in terms of $n = \log_2(N)$ (see [5] for details). By contrast, Shor's q-algorithm produces a probabilistic solution which is polynomial in n.

Set $r = \mathrm{ord}_N(a)$ and $n = \lceil \log_2(N) \rceil$. Next define the q-computation $U_a = U_{a,N}$ of order n by the relation

$$U_a|j\rangle = \begin{cases} |aj \bmod N\rangle & \text{if } j < N \\ |j\rangle & \text{if } N \leqslant j < 2^n. \end{cases}$$

It is indeed a q-computation, as the map $\mathbb{Z}_N \to \mathbb{Z}_N$ such that $j \mapsto aj \bmod N$ is bijective (a permutation map). The inverse q-computation is $U_{a^{-1},N}$. Finally define, for every $s \in \{0, \ldots, r-1\}$, the q-vector of order n

$$u_s = \sum_{j=0}^{r-1} e^{-2\pi i j \frac{s}{r}} |a^j \bmod N\rangle.$$

Applying the operator U_a to u_s we get

$$U_a u_s = \sum_{j=0}^{r-1} e^{-2\pi i j \frac{s}{r}} |a^{j+1} \bmod N\rangle = e^{2\pi i \frac{s}{r}} u_s,$$

which means that u_s *is an eigenvector of* $U_{a,N}$ *with eigenvalue* $e^{2\pi i \frac{s}{r}}$.

At this point it would seem natural to apply Kitaev's q-algorithm to estimate the phase s/r of $e^{2\pi i \frac{s}{r}}$, with the idea that the information gained in this way could give us precious information about r. However this does not work, since *the eigenvector u_s would be known only if r were already known*.

Fortunately this can be circumvented with the observation that

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1_n\rangle.$$

Indeed, if in Kitaev's q-algorithm we set $m = 2n + 1 + \lceil 2 + \frac{1}{2\varepsilon} \rceil$ and we let the initial state be $|0_m\rangle|1_n\rangle$, then, with probability $(1-\varepsilon)/r$, we will get an estimate $\tilde{\varphi} \approx s/r$ with $2n+1$ correct bits. Now we have that

$$\left|\frac{s}{r} - \tilde{\varphi}\right| \leqslant \frac{1}{2^{2n+1}} \leqslant \frac{1}{2r^2}$$

and hence, letting $s/r = s'/r'$ with $(s', r') = 1$, the inequality

$$\left|\frac{s'}{r'} - \widetilde{\varphi}\right| \leqslant \frac{1}{2r'^2}$$

also holds. By a well known result in continued fractions (see [6]), s'/r' must be a convergent of $\widetilde{\varphi}$. As $\widetilde{\varphi}$ is a rational number, its set of convergents is finite and can be computed by the continued fraction algorithm.

Summarizing, the choice of m in the phase estimation procedure assures that, with a probability of $1-\varepsilon$, there exists a convergent $\widetilde{\varphi}$ such that its denominator is either r if $(s, r) = 1$ or a divisor of r if $(s, r) \neq 1$.

If $(s, r) = 1$ then r is the order of a. This fact can be checked directly computing $a^{r_n} \mod N$ where s_n/r_n is a convergent of $\widetilde{\varphi}$. If $(s, r) \neq 1$, then $a^r \mod N$ is not equal to 1, and we need to repeat the phase estimation algorithm in order to get an estimation such that $(s, r) = 1$. Using the prime number theorem (see [5]), one can show that repeating the algorithm $O(n)$ times, with high probability we get an estimation $\widetilde{\varphi}$ with a convergent s/r such that $(s, r) = 1$ Details.

The number of steps of the whole q-algorithm is $O\left(n^4\right)$: the more complex step is associated to the continued fraction algorithm, with complexity $O\left(n^3\right)$, which needs to be repeated $O(n)$ times in order to assure, with high probability, a convergent s/r such that $(s, r) = 1$.

With further improvements of these ideas (see [7]) the complexity can be reduced to $O\left(n^3\right)$.

Let $1 < a < N$ be positive integers such that $(a, N) = 1$ and $\varepsilon > 0$ a (small) real number. The algorithm described below finds $r = \mathrm{ord}_N(a)$ with probability $1 - \varepsilon$ with an average number of iterations which is $O(n)$. The total complexity is $O(n^4)$. The algorithm ContFrac returns, given a rational number, the list of the denominators of its convergents. See ContFrac.

$\textsc{Shor-Order}[a, N, \varepsilon]$

$n = \lceil \log_2(N) \rceil$, $m = 2n + 1 + \log_2\left(2 + \frac{1}{2\varepsilon}\right)$

//Working q-space: $\mathbb{H}^{(m)} \otimes \mathbb{H}^{(n)}$

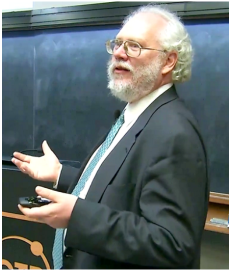
0. $\rightarrow |0_m\rangle|0_n\rangle$
1. $\textsc{Hadamard}[m]$ $\rightarrow \rho^m \sum_{j=0}^{2^m-1} |j\rangle|0_n\rangle$
2. $U_{a,N}$ $\rightarrow \frac{\rho^m}{\sqrt{r}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^m-1} e^{2\pi i j \frac{s}{r}} |j\rangle|u_s\rangle$
3. $\mathrm{QFT}^\dagger[m]$ $\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left|\widetilde{s/r}\right\rangle |u_s\rangle$
4. $M = M_{\{1,\ldots,n\}}$ $\rightarrow \widetilde{s/r}$
5. ContFrac $\rightarrow D$
6. for $r' \in D$ do

   if $a^{r'} \bmod N = 1$, return r'
7. return Not-successful

//$r'|r$, and $r' = r$ in $O(n)$ iterations ∎

Since the condition $r' = r$ is met in $O(n)$ iterations, we will get the correct order r with an average time $O(n^4)$. This is the algorithm we need in the next Section and will be denoted $\textsc{Shor-Order}(a, N)$.

Shor's factoring



Peter Shor

Let N be an odd positive integer that is not a prime power.

The main observation is that we can obtain a proper factor of N if we are able to produce an integer $x \in \{2, \ldots, N-1\}$ such that

1. $(x, N) = 1$;
2. $r = \mathrm{ord}_N(x)$ is even.
3. $x^{\frac{r}{2}} + 1$ or $x^{\frac{r}{2}} - 1$ is not divisible by N.

Indeed, since by definition r is the least positive integer such that $x^r \equiv 1 \mod N$ (the condition 1 implies that this number exists), we see that $x^r - 1 = \left(x^{\frac{r}{2}} - 1\right)\left(x^{\frac{r}{2}} + 1\right)$ is divisible by N. Then $\gcd\left(x^{\frac{r}{2}} - 1, N\right)$ are divisors of N, and at least one is a proper divisor of N.

```
In [1]: from PyM import *

In [2]: N = 868964876735596933
        x = 698131112366343466

In [3]: r=order(x,N)
        r, r//2

Out[3]: (1448274778685725 8, 724137389342862 9)
```

```
In [4]: X=power(x,(r//2),N)-1
        X

Out[4]: 431066552829123 88

In [5]: d1=gcd(X,N)
        d1

Out[5]: 102205879

In [6]: is_prime(d1)

Out[6]: True

In [7]: d2=N//d1
        d2

Out[7]: 850210267

In [8]: is_prime(d2)

Out[8]: True

In [9]: d1,d2

Out[9]: (102205879, 850210267)

In [10]: ifactor(N)

Out[10]: {850210267: 1, 102205879: 1}
```

Proposition Let N be a positive integer with $m \geqslant 2$ distinct prime factors. Then the density of the set

$$\{x \in \mathbb{Z}_N^* \mid r = \mathrm{ord}_N(x) \text{ is even and } x^{\frac{r}{2}} + 1 \text{ is not divisible by } N\}$$

in $\mathbb{Z}_N^*$ is $\geqslant 1 - \frac{1}{2^{m-1}}$. Hints

SHOR-FACTOR[N]

x, r, d
1. random(N) $\rightarrow x$
2. if $d = (x, N) > 1$: return d
3. SHOR-ORDER(x, N) $\rightarrow r$
4. if $r \equiv 1 \bmod 2$, go to 1.
5. if $d = \left(x^{\frac{r}{2}} - 1, N\right) > 1$ and $d < N$: return d
6. if $d = \left(x^{\frac{r}{2}} + 1, N\right) > 1$ and $d < N$: return d
7. go to 1 ■

The complexity of SHOR-FACTOR is determined by step 3, and so its average cost is $O(n^4)$, $n = \log_2(N)$.

A more detailed analysis shows that the average number of **go to** in steps 4 and 7 is $O(1)$ Comments

Appendix A

Remarks and Proofs

The number of Segre 2×2 determinants is $2^{2n-3} - 2^{n-2}$. The minimum number of sufficient conditions turns out to be $2^n - n - 1$ and for $n \geqslant 2$, $2^{2n-3} - 2^{n-2} \geqslant 2^n - n - 1$, with equality ($= 1$) only for $n = 2$. For $n = 3$, the values are 6 and 4 (so 2 redundant equations); for $n = 4$, 28 and 11, so 17 redundant equations; and for large n, the number of redundant equations is asymptotically equal to the number of equations. For $n = 10$, for example, the two numbers are 130816 and 1013, which means 129803 redundancies.

P

Let $U = [u_{jk}] \in \boldsymbol{U}^{(n)}$ and set $N = 2^n$. Then $U = e^{i\alpha} U_1 U_2 \cdots U_{N-1}$, with $\alpha \in \mathbb{R}$ and where $U_l = U_{l,l+1} \cdots U_{l,N}$, with $U_{l,j}$ an element of $\boldsymbol{U}^{(1)}$ acting on the plane $[|l\rangle, |j\rangle]$ in the standard form (using the reference $|j\rangle$ and $|k\rangle$) and acting as the identity on any $|k\rangle$ such that $k \neq l, j$. This expression of U can be constructed as follows. The matrix $U_{1,2}$ is taken as the identity if $u_{21} = 0$ and otherwise as

$$\begin{bmatrix} u_{11}/\rho & -\bar{u}_{21}/\rho \\ u_{21}/\rho & \bar{u}_{11}/\rho \end{bmatrix}, \ \rho = \sqrt{|u_{11}|^2 + |u_{12}|^2},$$

so that the entry 21 of the matrix $U_{1,2}^\dagger U$ is 0. Defining $U_{1,3}$, ..., $U_{1,N}$ in a similar way, we achieve that all entries of the first column of $U' = U_{1,N}^\dagger \cdots U_{1,2}^\dagger U$, other than the entry 11, are 0. Since U' is unitary, so that any two of its columns are orthogonal, all entries of the first row of U', other than the entry 11, are also 0. Since the entry 11 of U' is a unit complex number, we see that there is $\alpha_1 \in \mathbb{R}$ such that $e^{-i\alpha_1} U_{1,N}^\dagger \cdots U_{1,2}^\dagger U$ has the form

$$\begin{bmatrix} 1 & \mathbf{0}_n \\ \mathbf{0}_n^\dagger & V \end{bmatrix}, \; V \in \boldsymbol{U}^{(N-1)}.$$

Now, by induction, $V = e^{i\beta} U_2 \cdots U_{N-1}$, with $\beta \in \mathbb{R}$ and where $U_l = U_{l,l+1} \cdots U_{l,N}$ with $U_{l,j}$ an element of $\boldsymbol{U}^{(1)}$ acting on the plane $[|l\rangle, |j\rangle]$ and as the identity on any $|k\rangle$, $k \neq l, j$. Finally the claim follows by defining $\alpha = \alpha_1 + \beta$ and $U_1 = U_{1,2} U_{1,3} \cdots U_{1,N}$. Note that the number of the $U_{l,j}$ different from the identity is at most $N(N-1)/2$.

The proof can be completed on noticing that in the Example 42 we established that the $U_{l,j}$ can be expressed as a product of U-gates and $C_{r,s}$-gates.

P

We refer to Section 4.5.3 of [7] for a sketch of how the proof goes. But even in this encyclopedic book we read that providing all the details "is a little beyond our scope" (p. 198). A more complete proof, including the more subtle mathematical details, can be found in [8]. In particular it contains a full proof of the key fact that if $\cos\alpha = \cos^2(\pi/8)$, then α/π is irrational (Lemma 3.1.8).

P

Since $\mu_\xi = \frac{1}{N}\frac{N-M}{\sqrt{N-M}} = \sqrt{N-M}/N$,

$$\begin{aligned}
K(\xi) &= \sum_{\nu\in J_0}\left(2\sqrt{N-M}/N - 1/\sqrt{N-M}\right)|\nu\rangle + \sum_{\nu\in J_1}\frac{2\sqrt{N-M}}{N}|\nu\rangle \\
&= \sum_{\nu\in J_0}\frac{N-2M}{N\sqrt{N-M}}|\nu\rangle + \sum_{\nu\in J_1}\frac{2\sqrt{M}\sqrt{N-M}}{N\sqrt{M}}|\nu\rangle \\
&= \cos(\varphi)\xi + \sin(\varphi)\xi'.
\end{aligned}$$

Similarly, since $\mu_{\xi'} = M/N\sqrt{M} = \sqrt{M}/N$,

$$\begin{aligned}
K(\xi') &= \sum_{\nu\in J_0}\left(\frac{2\sqrt{M}}{N}\right)|\nu\rangle + \sum_{\nu\in J_1}\left(\frac{2\sqrt{M}}{N} - \frac{1}{\sqrt{M}}\right)|\nu\rangle \\
&= \sum_{\nu\in J_0}\left(\frac{2\sqrt{M}\sqrt{N-M}}{N}\frac{1}{\sqrt{N-M}}\right)|\nu\rangle + \sum_{\nu\in J_1}\left(\frac{2M-N}{N\sqrt{M}}\right)|\nu\rangle \\
&= \sin(\varphi)\xi - \cos(\varphi)\xi'.
\end{aligned}$$

P

The justification that GroverK computes the operator K follows from the following observations:

1. $K = 2P_{\boldsymbol{h}^{(n)}} - I_N$, where P_a denotes the orthogonal projector onto a (for unit a, $P_a(x) = \langle a|x\rangle a$). Indeed, the claim follows immediately from the relation

$$P_{\boldsymbol{h}^{(n)}}(x) = \langle \boldsymbol{h}^{(n)}|x\rangle \boldsymbol{h}^{(n)} = \rho^{2n}(\sum x_\nu)\sum|\nu\rangle = \mu_x \sum|\nu\rangle$$

and the definition of K (page 57).

2. $K = H^{\otimes n}\big(2P_{|0_n\rangle} - I_N\big)H^{\otimes n}$. This is a direct consequence of the formula $UP_aU^{-1} = P_{Ua}$, where U is a arbitrary q-computation and a any q-vector, and the preceding formula. Note that if we apply UP_aU^{-1} to Ux we obtain Ua if $x = a$ and 0 if x is orthogonal to a.

3. $I_N - 2P_{|0_n\rangle} = X^{\otimes n} C_{\{2,\dots,n\},1}(Z) X^{\otimes n}$. Note that $I_N - 2P_{|0_n\rangle}$ changes the sign of $|0_n\rangle$ and is the identity on all $|\nu\rangle$ with $\nu \neq 0_n$. In relation to the right hand side of the formula, observe that $C_{\{2,\dots,n\},1}(Z)$, and hence the whole composition, will do nothing on $|\nu\rangle$ if not all $\nu_2, \dots, \nu_n$ are 0. If $\nu_2 = \cdots \nu_n = 0$, then $C_{\{2,\dots,n\},1}(Z)$ applies Z to $|\bar{\nu}_1\rangle$, and, by the definition of Z ($Z|0\rangle = |0\rangle$, $Z|1\rangle = -|1\rangle$), this action does nothing if $\nu_1 = 1$ and changes its sign if $\nu_1 = 0$.

4. The analysis of Grover's *q*-algorithm has to be completed with a *q*-algorithm for $C_{\{2,\dots,n\},1}(Z)$. But this has been done in page 40.

P

The probability $p(|2^m\varphi - l| > 2^{m-r})$ is equal to

$$\sum_{l=-2^{m-1}+1}^{-(2^{m-r}+1)} p_l + \sum_{l=(2^{m-r}+1)}^{2^{m-1}} p_l .$$

Now the bound (∗∗), p. 83, can be derived from the explicit expression (∗) for p_l (page 82). We refer to [7] for further details.

P

The Prime Number Theorem asserts that the number of primes which are smaller than r is asymptotically equal to $\frac{r}{\log(r)}$. Hence, the probability of choosing (uniformly) a random prime number $0 < s < r$ is asymptotically equal to

$$p(0 < s < r, s \,\text{is prime}) = p \sim \frac{1}{\log r} > \frac{1}{\log N}.$$

Then the expected number of iterations in order to find a prime number $s < r$ is equal to:

$$\begin{aligned}\sum_{i=1}^{\infty} i(1-p)^{i-1} p &= p \sum_{i=1}^{\infty} i(1-p)^{i-1} \\ &= \frac{p}{(1-(1-p))^2} = \frac{1}{p} \sim \log(r) < \log(N).\end{aligned}$$

Hence, after not more than $\log(N) = O(n)$ choices, we expect to choose a value of s which is prime with r.

P

The continuous fraction representation of a rational number x is a vector of integers $[x_0, x_1, \ldots, x_n]$, with $x_j > 0$ for $j = 1, \ldots, n$. The relation between x and $[x_0, x_1, \ldots, x_n]$ can be displayed as a 'continuous fraction':

$$x = x_0 + \cfrac{1}{x_1 + \cfrac{1}{\ddots \, \cfrac{}{x_{n-1} + \cfrac{1}{x_n}}}}$$

By abuse of notation we will also write $x = [x_0, x_1, \ldots, x_n]$. In these terms the continuous fraction can be expressed by the recursive formula

$$[x_0, x_1, \ldots, x_n] = x_0 + \frac{1}{[x_1, \ldots, x_n]}$$

The rational numbers $c_j = [x_0, x_1, \ldots, x_j]$, $j = 0, 1, \ldots, n$, are called the *convergents* of the number x. The list of denominators $\{d_0, d_1, \ldots, d_n\}$ of these convergents can be computed recursively:

$$d_0 = 1, \quad d_1 = x_1, \quad d_j = x_j d_{j-1} + d_{j-2} \quad (j = 2, \ldots, n)$$

Actually it is easy to prove by induction that $c_j = m_j / d_j$, where

$$m_0 = x_0, \quad m_1 = x_1 x_0 + 1, \quad m_j = x_j m_{j-1} + m_{j-2} \quad (j = 2, \ldots, n)$$

Thus $\{d_0, d_1, \ldots, d_n\}$ can be computed as follows

```
ContFrac(x):=
a=floor(x), k=1, d={0,1}
while x!=a and j<n do
   x=1/(x-a)
   a=floor(x)
   d=d|{a*d.(j-1)+d.(j-2)}
   j=j+1
return tail(d)
```

P

Prove that

$$p\left(x \in \mathbb{Z}_N^* \,|\, r = \mathrm{ord}_N(x) \text{ is odd or } x^{\frac{r}{2}} + 1 \text{ is divisible by } N\right) \geqslant \frac{1}{2^m}.$$

For that, write $N = p_1^{\alpha_1} \dots p_m^{\alpha_m}$, where $p_1, \dots, p_m$ are distinct prime numbers. Then, $\mathbb{Z}_N^* = \mathbb{Z}^*_{p_1^{\alpha_1}} \times \cdots \times \mathbb{Z}^*_{p_m^{\alpha_m}}$. Write x_j for the reduction of $x \mod p_j^{\alpha_j}$, and r_j for the order of x_j in $\mathbb{Z}^*_{p_j^{\alpha_j}}$. Denote by d_j the biggest exponent such that 2^{d_j} divides r_j. Denote by d the biggest exponent such that 2^d divides r. Then, it is easy to show that if r is odd or if r is even and $x^{\frac{r}{2}} \equiv -1 \mod N$, then $d_j = d$ for all d.

To conclude, use that if 2^{d_j} is the largest power of 2 dividing $\varphi(p_j^{\alpha_j})$, then

$$p\left(x \in \mathbb{Z}_N^* \,|\, 2^{d_j} \text{ divides } \mathrm{ord}_{p_j^{\alpha_j}}(x)\right) = \frac{1}{2}.$$

P

Denote by p the probability of the event
$\{x \in \mathbb{Z}_N^* \mid r = \mathrm{ord}_N(x) \text{ is even and } x^{\frac{r}{2}} + 1 \text{ is not divisible by } N\}$,
when x is chosen uniformly at random at $\mathbb{Z}_N^*$. Then, the expected number of iterations of the algorithm is equal to:

$$\sum_{i=1}^{\infty} i(1-p)^{i-1}p = p\sum_{i=1}^{\infty} i(1-p)^{i-1} = \frac{1}{p} \leqslant \frac{2^{m-1}}{2^{m-1}-1} = 1+\frac{1}{2^{m-1}-1}.$$

As $m > 1$, the expected number of iterations is $O(1)$.

P

Appendix B

Physics footnotes

In order to execute q-programs of order n on a physical support, it is required to have a quantum register of length n capable of being initialized at any state, $|0_n\rangle$ by default, and "implementations" of the operations

- $R_j(U)$ [with $U \in \{H, S, T\}$ in the restricted case]
- $C_{j,k}$
- $M_L(\sigma)$ for any state σ and any subregister L. In particular, $M(\sigma)$ when L is the whole register. If $|L| = r$, $M_L(\sigma)$ delvers $M \in B^r$ with probability $p_M = |\sigma_L^M|^2$ and resets the state to $u(\sigma_L^M)$.

A *quantum computer* (of order n) is a quantum register $\Sigma^{(n)}$ endowed with such implementations.

Its main beauty is that such a computer allows us to perform (or approximate) any q-computation.

In its most basic form, the *no-cloning theorem* is the assertion that there is no q-computation U of order 2 that satisfies

$$U(|x\rangle|0\rangle) = |x\rangle|x\rangle,$$

for all one q-bit states x.

Indeed, consider $|x\rangle = \rho(|b\rangle + |b'\rangle)$, $b \in B$, and $b' = 1 + b$. Then we have

$$U(|x\rangle|0\rangle) = \begin{cases} |x\rangle|x\rangle = \rho^2(|b\rangle|b\rangle + |b\rangle|b'\rangle + |b'\rangle|b\rangle + |b'\rangle|b'\rangle) \\ \rho U(|b\rangle|0\rangle + |b'\rangle|0\rangle) = \rho(|b\rangle|b\rangle + |b'\rangle|b'\rangle) \end{cases},$$

which is a contradiction.

A possible state of a q-register of order 2 is

$$\sigma = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Let's assume that the first q-bit is at A and the other at B. If A and B successively measure their q-bit, it turns out that *they get the same result*.

Indeed, the state at A collapses in $|00\rangle$ or in $|11\rangle$ depending on whether A measures 0 or 1, respectively (i.e., the normalized orthogonal projection of σ in the space $\{|0b\rangle\}_{b\in B}$ is $|00\rangle$ and is $|11\rangle$ in the space $\{|1b\rangle\}_{b\in B}$. Thus the state of the pair is $|00\rangle$ if A measures 0 and it $|11\rangle$ if A measures 1. It is thus clear that the measurement of the second q-bit by B will be 0 in the first case and 1 in the second.

This situation puzzled its discoverers, Einstein, Podolski and Rolfsen (and anyone since then) because it appeared to them as a 'spooky action at a distance'.

Quantum computing techniques allow transferring the state of a q-bit in A to the same state of a q-bit in B (the state disappears in A and appears in B). Here is an outline of the procedure.

- Let $\sigma = \alpha|0\rangle + \beta|1\rangle$ be the (unknown) state of a q-bit in A that we wish to teleport to B.
- Let $\tau = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ an EPR state shared by A and B.
- A performs a C_{12} gate on the state

$$\sigma\tau = \frac{1}{\sqrt{2}}[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle)],$$

obtaining the state

$$\frac{1}{\sqrt{2}}[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle)].$$

- Now A performs H on the first q-bit and obtains

$$\frac{1}{\sqrt{2}}[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)].$$

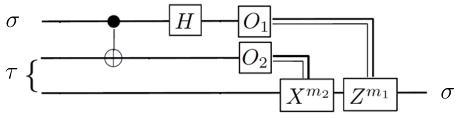
This can be rearranged in the form

$$\frac{1}{\sqrt{2}}\big(|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|0\rangle - \beta|1\rangle) +$$
$$|10\rangle(\alpha|1\rangle + \beta|0\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)\big)$$

Now A measures q-bits 1 and 2. The following table shows, for each of the possible results, the status of the q-bit in B:

Result	00	01	10	11
q-bit B	$\alpha\|0\rangle + \beta\|1\rangle$	$\alpha\|0\rangle - \beta\|1\rangle$	$\alpha\|1\rangle + \beta\|0\rangle$	$\alpha\|1\rangle - \beta\|0\rangle$
Action in B	I	X	Z	XZ

- Finally B can reproduce the state σ in its q-bit if it knows the result of the measurement made by A (00, 01, 10 or 11) by simply performing the actions I, X, Z or XZ, respectively.



Outlook

" Even if we don't have general purpose quantum computers, we have already expanded considerably our understanding of, among others, *quantum information theory*, *quantum cryptography*, *quantum Hamiltonian dynamics*, *classical computational complexity theory*, *the nature of randomness*, and basic issues at the heart of the philosophy of science—including whether quantum mechanics itself is a falsifiable theory. In short, quantum computing is pretty irresistible!" (from [9], AVI WIGDERSON's review of Aaronson's book *Quantum Computing since Democritus* [10]. And also this: "the book is not really about quantum computing. It is far broader and uses quantum computing as an opportunity to introduce a whole set of important concepts in math, physics, philosophy, and, above all, computational complexity theory".

Quantum Algorithm Zoo: [11].

[12]: *Lecture Notes on Quantum Algorithms for Scientific Computation*.

[13]: "... we then employ the QSVT [Quantum Singular Value Transformation] to construct intuitive quantum algorithms for search, phase estimation, and Hamiltonian simulation, and also showcase algorithms for the eigenvalue threshold problem and matrix inversion. This overview illustrates how the QSVT is a single framework comprising the three major quantum algorithms, suggesting a grand unification of quantum algorithms."

[14]: *Quantum Computing: Lecture Notes*. It contains 252 references, up to 2022.

References I

[1] D. Aerts and I. Daubechies, “Physical justification for using the tensor product to describe two quantum systems as one joint system,” *Helvetica Physica Acta*, vol. 51, no. 5-6, 1978.

[2] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219, 1996.

[3] L. K. Grover, “From Schrödinger’s equation to quantum search algorithm,” *American Journal of Physics*, vol. 69, no. 7, pp. 769–777, 2001.

[4] A. Y. Kitaev, “Quantum measurements and the abelian stabilizer problem,” *Electr. Coll. Comput. Complex.*, vol. 3, p. 22 pp., 1995.

References II

[5] T. Apostol, *Introduction to Analytic Number Theory*.

Undergraduate Texts in Mathematics, Springer-Verlag, 1976.

[6] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*.

Oxford University Press, 2008 (6th edition, revised by D. R. Heath-Brown and J. H. Silvermann; 1st edition published in 1938).

[7] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*.

Cambridge University Press, 2000 (5th printing 2005).

References III

[8] K. R. Parthasarathy, *Lectures on Quantum Computation, Quantum Error Correcting Codes and Information Theory*.

Narosa Publishing House, 2006.

(For the Tata Institute of Fundamental Research, international distribution by AMS).

[9] A. Wigderson, "Review of "Quantum Computing Since Democritus" *(Aaronson 2013)*," 2014.

https://math.ipm.ac.ir/combin/useful_material/rnoti-p1218.pdf.

[10] S. Aaronson, *Quantum computing since Democritus*.

Cambridge University Press, 2013.

References IV

[11] S. P. J. Jordan, "Quantum Algorithm Zoo," 2023.
Web page. Consulted 28/06/2023.

[12] L. Lin, "Lecture notes on quantum algorithms for scientific computation," 2022.
arXiv pdf.

[13] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, "Grand unification of quantum algorithms," *PRX Quantum*, vol. 2, no. 4, p. 040203, 2021.

[14] R. De Wolf, "Quantum computing: Lecture notes *(v5)*," 2023.
arXiv pdf.