УДК 512.6, 517.9, 519.6

SOME INTERRELATIONS IN MATRIX THEORY AND LINEAR ALGEBRA FOR QUANTUM COMPUTING AND QUANTUM ALGORITHMS DESIGN. PT.2

Reshetnikov Andrey[1], Tyatyushkina Olga[2], Ulyanov Sergey[3], Tanaka Takayuki[4], Rizzotto Giovanni[5]

[1]*PhD in informatics, associate professor;*
Dubna State University,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: agreshetnikov@gmail.com.

[2]*PhD, associate professor;*
Dubna State University,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: tyatushkina@mail.ru.

[3]*Doctor of Science in Physics and Mathematics, professor;*
Dubna State University,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail:ulyanovsv@mail.ru.

[4]*PhD, professor;*
The Graduate School of Information Science and Technology, Hokkaido University;
N14, W9, Sapporo-shi, Hokkaido, Japan;
e-mail:ttanaka@ssc.ssi.ist.hokudai.ac.jp.

[5]*PhD, professor;*
Corporate Advanced System, ST Microelectronics,
Via C. Olivetti 2, 20041 Agrate Brianze, Italy;
e-mail: gda@dsi.unimi.it.

The goal of this pedagogical article is to describe for IT engineering researchers and master course students main mathematical operations with matrices and linear operators used in quantum computing and quantum information theory. These operations are defined in the framework of linear algebra and therefore for their understanding there is no need to introduce physical background of quantum mechanics.

Keyword: matrix theory, linear algebra, quantum computing, quantum information, quantum algorithm.

НЕКОТОРЫЕ ВЗАИМОСВЯЗИ В МАТРИЧНОЙ ТЕОРИИ И ЛИНЕЙНОЙ АЛГЕБРЕ ДЛЯ КВАНТОВЫХ ВЫЧИСЛЕНИЙ И ПРОЕКТИРОВАНИЯ КВАНТОВЫХ АЛГОРИТМОВ. Ч.2.

Решетников Андрей Геннадьевич[1], Тятюшкина Ольга Юрьевна[2], Ульянов Сергей Викторович[3], Танака Такаюки[4], Риззотто Джиованни[5]

[1]*Доктор информатики (PhD in Informatics), к.т.н., доцент;*
ГБОУ ВО МО «Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: agreshetnikov@gmail.com.

[2]*Кандидат технических наук, доцент;*
ГБОУ ВО МО «Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: tyatushkina@mail.ru.

[3]*Доктор физико-математических наук, профессор;*
ГБОУ ВО МО «Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail:ulyanovsv@mail.ru.

[4]*Доктор наук (PhD in Informatics),*
Высшая школа информатики и технологии,
Университет Хоккайдо;
N14, W9, Саппоро-Ши, Хоккайдо, Япония141980;
e-mail: ttanaka@ssc.ssi.ist.hokudai.ac.jp.

[5]*Доктор информатики (PhD in Informatics);*
Корпоративные системы, ST Microelectronics,
Via C. Olivetti 2, 20041 Agrate Brianze, Италия;
e-mail: gda@dsi.unimi.it.

Цель этой педагогической статьи - описать для ИТ-инженеров и магистрантов основные математические операции с матрицами и линейными операторами, используемыми в квантовых вычислениях и квантовой теории информации. Эти операции определены в рамках линейной алгебры, и поэтому для их понимания нет необходимости вводить физические основы квантовой механики.

Ключевые слова: Теория матриц, линейная алгебра, квантовые вычисления, квантовая информация, квантовый алгоритм

Introduction

Matrices have been the subject of much study, and large bodies of results have been obtained about them. In this article we introduce main definitions of matrix theory and study the interplay between the theory of matrices and the theory of orthogonal polynomials in quantum computing [1-7]. Interesting results have been obtained for Krawtchouk polynomials and also for generalized Krawtchouk polynomials. More recently, it was obtained conditions for the existence of integral zeros of binary Krawtchouk polynomials. Also it was obtained properties for generalized Krawtchouk polynomials. Other generalizations of binary Krawtchouk polynomials have also been considered. Generalized some properties of binary Krawtchouk polynomials to q-Krawtchouk polynomials, derived orthogonality relations for quantum and q-Krawtchouk polynomials and showed that affine q-Krawtchouk polynomials are dual to quantum q-Krawtchouk polynomials. In this paper, we define and study a generalization of Krawtchouk polynomials, namely, m-polynomials [8-11]. Applications of Krawtchouk / Hadamard matrices in quantum algorithm's design are considered.

Krawtchouk matrices and quantum random walk

In quantum probability, random variables are modelled by self-adjoint operators on Hilbert space and independence by tensor products. We can model a symmetric Bernoulli random walk as follows. Consider a 2-dimensional Hilbert space $V = \boldsymbol{R}^2$ and two special 2×2 operators

$$F = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad G = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

satisfying $F^2 = G^2 = I$, the 2×2 identity. Recall the fundamental Krawtchouk/Hadamard matrix H (see, Part 1, Eqs (2) and (4)), which we shall now view as

$$H = F + G = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

One can readily check that

$$FH = F\left(F+G\right) = \left(F+G\right)G = HG \tag{1}$$

(use $F^2 = G^2 = I$). This, of course, can be viewed as the spectral decomposition of F and we can interpret the Hadamard matrix as diagonalizing F .

Remark. Note that the exponentiated operator

$$\exp\left(zF\right) = \begin{pmatrix} \cosh z & \sinh z \\ \sinh z & \cosh z \end{pmatrix}$$

has the expectation value in the state e_0 equal

$$\left\langle e_0, \exp\left(zF\right)e_0 \right\rangle = \cosh z$$

which coincides with the moment generating function for the symmetric Bernoulli random variable taking values ± 1. This shows that indeed we are dealing with the (quantum) generalization of the classical model.

Remark. **(***Quantum computing interpretation***).** We can view $V = \operatorname{span}\left\{e_0, e_1\right\}$ as a Hilbert space representing a simple two-state quantum system, e.g., a particle with possible spin-up (e_0) and spin-down (e_1) states, respectively. In the context of "quantum computing", these elementary states represent "qubits". The operator F represents a "spin flip" and G a "phase change", both are thus basic qubit operations. Geometrically, they are reflections in V through lines that form 45 degrees. The Hadamard matrix M is essentially a "rotation by 45 degrees" (up to composition with reflection and scaling) and therefore ties them via (1). Now, to perform quantum computing, one needs an assembly of N such independent systems – this constitutes a simple quantum computer. The Hilbert space of computer states is represented by the N -th tensor product of the original space V , that is, by the 2^N -dimensional Hilbert space $V^{\otimes N}$. This motivates our further considerations.

Define the following simple operators (cf. Appendix 1)

$$f_1 = F \otimes I \otimes \cdots \otimes I$$

$$f_2 = I \otimes F \otimes I \otimes \cdots \otimes I$$

$$\vdots = \vdots$$

$$f_N = I \otimes I \otimes \cdots \otimes F$$

each f_i describing a "flip" at the i -th position. These are our quantum equivalents of the random walk *variables*. Performing them independently would result in "running" a classical TV-bit computer. We shall consider the superposition of these independent actions, setting

$$X_F = f_1 + \cdots + f_N .$$

Notation: For notational clarity, since N is fixed throughout the discussion, we drop N indices on X 's.

Analogously, we define:

$$g_1 = G \otimes I \otimes \cdots \otimes I$$

$$g_2 = I \otimes G \otimes I \otimes \cdots \otimes I$$

$$\vdots = \vdots$$

$$g_N = I \otimes I \otimes \cdots \otimes G$$

With $X_G = g_1 + \cdots + g_N$. We also extend H to the N -fold tensor product, setting $H_N = H^{\otimes N}$. (This is the same as $H^{(N)}$ of the previous sections.)

For illustration, consider a calculation for $N = 3$:

$$f_1 H_3 = (F \otimes I \otimes I)(H \otimes H \otimes H)$$

$$= (H \otimes H \otimes H)(G \otimes I \otimes I) = H_3 g_1$$

where the relation $FH = HG$ is used. This clearly generalizes to $f_k H_N = H_N g_k$ and, by summing over k, yields an important relation:

$$X_F H_N = H_N X_G .$$

Since products are preserved when reducing to the symmetric tensor space, we get

$$\bar{X}_F \bar{H}_N = \bar{H}_N \bar{X}_G$$

the bars indicating the corresponding induced maps (see Appendix 2). We know how to calculate $\bar{H}_N$ from the action of H on polynomials in degree N. Note that for symmetric tensors we have the components $x_0^{N-k} x_1^k$ in degree N for $0 \le k \le N$.

Proposition. For each $N > 0$, symmetric reduction of Hadamard matrices leads to $\bar{H}_{ij} = \Phi_{ij}^{(N)}$ That is, $\bar{H}_N$ is the transpose of the Krawtchouk matrix $\Phi^{(N)}$.

Proof: Writing (x, y) for (x_0, x_1), we have in degree N for the k th component:

$$(x+y)^{N-k}(x-y)^k = \sum_l \bar{H}_{kl} x^{N-l} y^l .$$

Scaling out x^N and replacing $\upsilon = y/x$ yields the generating function for the Krawtchouk matrices with the coefficient of υ^l equal to $\Phi_{lk}^{(N)}$. Thus the result.

Now consider the generating function for the elementary symmetric functions in the quantum variables f_j. This is the N - fold tensor power

$$\mathcal{F}_N(t) = (I + tF)^{\otimes N} = I^{\otimes N} + tX_F + \cdots$$

noting that the coefficient of t is X_F. Similarly, define

$$\mathcal{G}_N(t) = (I + tG)^{\otimes N} = I^{\otimes N} + tX_G + \cdots$$

From $(I + tF)H = H(I + tG)$ we have

$$\mathcal{F}_N H_N = H_N \mathcal{G}_N \text{ and } \bar{\mathcal{F}}_N \bar{H}_N = \bar{H}_N \bar{\mathcal{G}}_N .$$

The difficulty is to calculate the action on the symmetric tensors for operators, such as X_F, that are not pure tensor powers. However, from $\mathcal{F}_N(t)$ and $\mathcal{G}_N(t)$ we can recover X_F and X_G via

$$X_F = \frac{d}{dt}\bigg|_{t=0} (I + tF)^{\otimes N}, \quad X_G = \frac{d}{dt}\bigg|_{t=0} (I + tG)^{\otimes N}$$

with corresponding relations for the barred operators. Calculating on polynomials yields the desired results as follows:

$$I+tF=\begin{pmatrix}1 & t\\ t & 1\end{pmatrix},\qquad I+tG=\begin{pmatrix}1+t & 0\\ 0 & 1-t\end{pmatrix}.$$

In degree N, using x and y as variables, we get the k th component for $\bar{X}_F$ and $\bar{X}_G$ via

$$\left.\frac{d}{dt}\right|_{t=0}\left(x+ty\right)^{N-k}\left(tx-y\right)^{k}=\left(N-k\right)x^{N-(k+1)}y^{k+1}+kx^{N-(k-1)}y^{k-1}$$

and since $I+tG$ is diagonal,

$$\left.\frac{d}{dt}\right|_{t=0}\left(1+t\right)^{N-k}\left(1-t\right)^{k}x^{N-k}y^{k}=\left(N-2k\right)x^{N-k}y^{k}$$

For example, calculation for $N=4$ result in

$$\bar{X}_F=\begin{pmatrix}0 & 4 & 0 & 0 & 0\\ 1 & 0 & 3 & 0 & 0\\ 0 & 2 & 0 & 2 & 0\\ 0 & 0 & 3 & 0 & 1\\ 0 & 0 & 0 & 4 & 0\end{pmatrix},\qquad \bar{H}_4=\begin{pmatrix}1 & 4 & 6 & 4 & 1\\ 1 & 2 & 0 & -2 & -1\\ 1 & 0 & -2 & 0 & 1\\ 1 & -2 & 0 & 2 & -1\\ 1 & -4 & 6 & -4 & 1\end{pmatrix},$$

$$\bar{X}_G=\begin{pmatrix}4 & 0 & 0 & 0 & 0\\ 0 & 2 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & -2 & 0\\ 0 & 0 & 0 & 0 & -4\end{pmatrix}.$$

We observe the spectrum of $\bar{X}_N$ is $N,N-2,\ldots,2-N,-N$, which coincides with the support of the classical random walk.

We note that the top row of $\left(I+tF\right)^{\otimes N}$ is $t^{w(k)}$ where $w(k)$ is the binary shuffling function of section 3. This is seen by noting that each time one tensors with $I+tF$, the original top row is reproduced then it is concatenated with a replica of itself modified in that each entry picks up a factor of t. Now, collapsing to the symmetric tensor space, the top row will have entries $\binom{N}{k}t^{k}$. This follows as well by direct calculation of the 0th component matrix elements in degree N, namely by expanding $\left(x+ty\right)^{N}$.

To find the distributions, we must calculate expectation values. In the present context, expectation values in two particular states are especially interesting. Namely, in the state e_0 and the normalized trace – the uniform distribution on the spectrum. In the N -fold tensor product, we want to consider expectation values in the "ground state" $\left|000\ldots0\right\rangle$ and normalized traces. Then we can go to the symmetric tensors. Since everything factors, one easily obtains the expectation value of $\exp\left(zX_N\right)$ in the ground state $\left|000\ldots0\right\rangle$ to

be $(\cosh z)^N$. For the trace, tr $\exp(zF) = 2\cosh z$ implies $\operatorname{tr}\exp(zX_F) = 2^N(\cosh z)^N$ and, after normalizing, this yields $(\cosh z)^N$.

For the barred operators, we consider the symmetric trace. Here we can use the symmetric trace theorem, detailed in Appendix 2. It tells us that the generating function for the symmetric traces of any operator A in the various degrees is $\det(I - tA)^{-1}$. Taking $A = \exp(zF)$, we have

$$\det\left(I - te^{zF}\right)^{-1} = \left[\left(1 - te^{z}\right)\left(1 - te^{-z}\right)\right]^{-1}$$

$$= \left(1 - 2t\cosh z + t^2\right)^{-1}.$$

The latter is the generating function for Chebyshev polynomials of the second kind, so that the normalized symmetric trace is

$$(N+1)^{-1}\operatorname{tr}_{Sym}^{N}\exp(zF) = U_N(\cosh z)/(N+1).$$

This equals as well

$$\frac{e^{z(N+1)} - e^{-z(N+1)}}{\left(e^{z} - e^{-z}\right)(N+1)}$$

and that completes this study.

If A is represented by a matrix, given the matrix form $A_N = A^{\otimes N}$ computed as an N-fold Kronecker product, to reduce to $\overline{A}_N$, we see that acting on polynomials, for a fixed row label in the full tensor space, the column entries corresponding to basic tensors equivalent under permutation are summed to a single column. Then the matrix elements are chosen, one row from each equivalence class of basic tensors. What if $d = 2?$ Then row and column labels are single indices so that $\overline{A}_N$ is an $(N+1)\times(N+1)$ matrix with labels according to the exponent of x_1. That is, the basis for polynomials in degree N is given by the monomials $x_0^{N-k}x_1^k$, $0 \le k \le N$. The binary shuffling function and contraction operations are exactly the reduction to symmetric tensors and then induced matrix, (see Table 1 and 2) respectively.

Table 1: Krawtchouk matrices

$$\Phi^{(0)} = [1]$$

$$\Phi^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \Phi^{(2)} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix}$$

$$\Phi^{(3)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 1 & -1 & -3 \\ 3 & -1 & -1 & 3 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad \Phi^{(4)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 4 & 2 & 0 & -2 & -4 \\ 6 & 0 & -2 & 0 & 6 \\ 4 & -2 & 0 & 2 & -4 \\ 1 & -1 & 1 & -1 & 1 \end{bmatrix}$$

$$\Phi^{(5)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 3 & 1 & -1 & -3 & -5 \\ 10 & 2 & -2 & -2 & 2 & 10 \\ 10 & -2 & -2 & 2 & 2 & -10 \\ 5 & -3 & 1 & 1 & -3 & 5 \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

$$\Phi^{(6)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 6 & 4 & 2 & 0 & -2 & -4 & -6 \\ 15 & 5 & -1 & -3 & -1 & 5 & 15 \\ 20 & 0 & -4 & 0 & 4 & 0 & -20 \\ 15 & -5 & -1 & 3 & -1 & -5 & 15 \\ 6 & -4 & 2 & 0 & -2 & 4 & -6 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix}$$

Table 2: Symmetric Krawtchouk matrices

$$S^{(0)} = \begin{bmatrix} 1 \end{bmatrix}$$

$$S^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$S^{(2)} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$$S^{(3)} = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 3 & -3 & -3 \\ 3 & -3 & -3 & 3 \\ 1 & -3 & 3 & -1 \end{bmatrix}$$

$$S^{(4)} = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 0 & -12 & 0 & 6 \\ 4 & -8 & 0 & 8 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

$$S^{(5)} = \begin{bmatrix} 1 & 5 & 10 & 10 & 5 & 1 \\ 5 & 15 & 10 & -10 & -15 & -5 \\ 10 & 10 & -20 & -20 & 10 & 10 \\ 10 & -10 & -20 & 20 & 10 & -10 \\ 5 & -15 & 10 & 10 & -15 & 5 \\ 1 & -5 & 10 & -10 & 5 & -1 \end{bmatrix}$$

$$S^{(6)} = \begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 6 & 24 & 30 & 0 & -30 & -24 & -6 \\ 15 & 30 & -15 & -60 & -15 & 30 & 15 \\ 20 & 0 & -60 & 0 & 60 & 0 & -20 \\ 15 & -30 & -15 & 60 & -15 & -30 & 15 \\ 6 & -24 & 30 & 0 & -30 & 24 & -6 \\ 1 & -6 & 15 & -20 & 15 & -6 & 1 \end{bmatrix}$$

Let us consider the relationship between the structure of Simon's algorithm and Walsh transforms.

Entropy and Hadamard matrices

We will define the entropy of an orthogonal matrix. It provides a new interpretation of Hadamard matrices as those that saturate the bound for entropy. This definition play important role in QAs simulation, while the Hadamard matrix is used for preparation of superposition states and in entanglement-free QAs. We define the entropy of orthogonal matrices and Hadamard matrices (appropriately normalized) saturate the bound for the maximum of the entropy. The maxima (and other saddle points0 of the entropy function have an intriguing structure and yield generalizations of Hadamard matrices.

Consider n random variables with a set of possible outcomes $i = 1,\ldots,n$ having probabilities p_i, $i = 1,\ldots,n$. We have $\sum_{i=1}^{n} p_i = 1$ and the Shannon entropy $S^{Sh}\left(p_i\right) = -\sum_{i=1}^{n} p_i \ln p_i$.

We now define entropy of an orthogonal matrix O^i_j, $i, j = 1,\ldots,n$. Here O^i_j are real numbers with the constraint $\sum_{i=1}^{n} O^i_j O^i_k = \delta_{jk}$. In particular, the j th row of the matrix is a normalized vector for each $i = 1,\ldots,n$. We may associate probabilities $p^{(i)}_j = \left(O^i_j\right)^2$ with the i-th row, as $\sum_{j=1}^{n} p^{(i)}_j = 1$ for each i. We define the Shannon entropy for the orthogonal matrix as the sum of the entropies for each row:

$$S^{Sh}\left(O^i_j\right) = -\sum_{i,j=1}^{n} \left(O^i_j\right)^2 \ln\left(O^i_j\right)^2 .$$

The minimum value zero is attained by the identity matrix $O^i_j = \delta^i_j$ and related matrices obtained by interchanging rows or changing the signs of the elements. The entropy of the i-th row can have the maximum value $\ln n$, which is attained when each element of the row is $\pm\frac{1}{\sqrt{n}}$. This gives the bound, $S^{Sh}\left(O^i_j\right) \le n \ln n$.

In general, the entropy of an orthogonal matrix cannot attain this bound because of the orthogonality constraint $\sum_{i=1}^{n} O^i_j O^i_k = \delta_{jk}$, which constraints $p^{(i)}_j$ for different rows. In fact the bound is obtained only by the Hadamard matrices (rescaled by $\frac{1}{\sqrt{n}}$). Thus we have the criterion for the Hadamard matrices (appropriately normalized): those orthogonal matrices which saturate the bound for entropy.

Remark. The entropy is large when each element is as close to $\pm\frac{1}{\sqrt{n}}$ possible, i.e., to a main diagonal. Thus maximum entropy is similar to the maximum determinant condition of the Hadamard. The peaks of the entropy are isolated and sharp in contrast to the determinant.

Example. Matrix that maximizing the entropy for $n=3$ is

$$\begin{pmatrix} -\frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & -\frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & -\frac{1}{3} \end{pmatrix}.$$

For $n=5$, the result is similar: the magnitudes of the elements in each row are $\frac{2}{5}$ repeated 4 times and a diagonal element is a $-\frac{3}{5}$. This set can be generalized for any n. The matrix with $-\frac{n-2}{n}$ along the diagonal and each off-diagonal as $\frac{2}{n}$ is orthogonal. Each row is normalized as a consequence of the identity: $n^2=(n-2)^2+2^2(n-1)$.

For each n, there are saddle points apart from maxima and minima.

Example. For $n=3$ there is a saddle point and the corresponding matrix is

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{2} & -\frac{1}{\sqrt{2}} & \frac{1}{2} \end{pmatrix}.$$

The entropy peaks quite sharply at all extrema. Thus the entropy has a rich set of sharp extrema.

This result shows the important role of Hadamard operator in entanglement-free QA: with Hadamard transformation it is possible introduce maximal hidden information about classical basis independent states and superposition includes this maximal information. Thus, with superposition operator is possible created a new QA without entanglement while in any cases superposition includes information about the property of function f.

General properties of Walsh-Hadamard transformation $W(a,b,q)$. (see Appendix 3)

Let us consider a function $W:\mathcal{N}\times\mathcal{N}\times\left\{2^n\middle|n\in\mathbb{Z}^+\right\}\to\{-1,1\}$. The function satisfies the following conditions:

$W(0,b,q)=1$	$W(a\oplus c,b,q)=W(a,b,q)W(c,b,q)$
$W(a,b,q)=W(b,a,q)$	$\sum_{c=o}^{q-1}W(a,c,q)W(c,b,q)=q\delta_{ab}$

where the operator $\oplus$ is a bitwise exclusive-OR (XOR). Moreover, we define the transform W_q as follows: $|a\rangle \xrightarrow{W_q} \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} W(a,c,q)|c\rangle$. When the a-th row and b-th column element of a matrix U_{W_q} is $\frac{1}{\sqrt{q}} W(a,b,q)$ (unless otherwise specified, the rows and columns will be indexed beginning with 0), the matrix is a unitary matrix because of the definition of $W(a,b,q)$. Thus, the Quantum Turing Machine (QTM) can execute the transform W_q.

Example. Let $W(a,b,q=2^n) = (-1)^{a \cdot b}$ where $a \cdot b$ is an inner product of a and b, i.e., for $a = \sum_{i=0}^{n-1} a_i 2^i$ and $b = \sum_{i=0}^{n-1} b_i 2^i \quad a_i, b_i \in \{0,1\}, \quad a \cdot b = \sum_{i=0}^{n-1} a_i b_i \pmod 2$. Obviously, the function $W(a,b,2^n)$ satisfies the conditions above of $W(a,b,q)$. In fact, by using $W(a,b,2^n)$, in the Simon algorithm $W(s,c,2^n) = 1, s \cdot c = 0$.

Next, let us consider the case when $W(a,b,q)$ is a discrete Walsh function. We will show that the function $W(a,b,2^n)$ is a kind of discrete Walsh function.

First, we describe Walsh functions and discrete Walsh functions. We define a function

$$r(x): \mathbb{R} \to \{-1,1\} \text{ by } r(x) = \begin{cases} 1, & 0 \le x < \frac{1}{2} \\ -1, & \frac{1}{2} \le x < 1 \end{cases}$$

and $r(x+1) = r(x)$. Moreover, let $r_l(x)$ be a function $r_l(x) = r(2^l x)$, where $x \in \mathbb{R}, l \in \mathbb{N}$. Then, a *Walsh function* (more precisely, *Walsh-Paley function*) $W_l(x)$ is defined by $W_l(x) = \prod_{k=0}^{\infty} r_k(x)^{l_k}$, where $l = \sum_{k=0}^{\infty} l_k 2^k, l_k \in \{0,1\}$.

Remark. Every value of the function $W_l(x)$ is always a finite value because $l_k = 0$ for k that is sufficiently large (in fact, each value of the Walsh-Paley function is either 1 or -1). Moreover, when let $a \in \mathbb{N}$, $b \in \mathbb{N}$, and $q \in \mathbb{Z}^+$, a *discrete Walsh-Paley function* is defined by $W_a\left(\frac{b}{q}\right)$.

Example. Now, let $q = 2^n \ (n \in \mathbb{N})$. Then, the function $W_a\left(\frac{b}{q}\right)$ satisfies the following properties [so that this function satisfies the properties of $W(a,b,q)$]:

$W_0\left(\frac{b}{q}\right) = 1$	$W_{a \oplus c}\left(\frac{b}{q}\right) = W_a\left(\frac{b}{q}\right) W_c\left(\frac{b}{q}\right)$
$W_a\left(\frac{b}{q}\right) = W_b\left(\frac{a}{q}\right)$	$\sum_{c=o}^{q-1} W_a\left(\frac{c}{q}\right) W_c\left(\frac{b}{q}\right) = q\delta_{ab}$

Namely, a matrix U_{P_q} whose the a-th row and b-th column element is $\frac{1}{\sqrt{q}} W_a\left(\frac{b}{q}\right)$ is a unitary matrix. The transform $P_q : |a\rangle \xrightarrow{P_q} \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} W_a\left(\frac{c}{q}\right)|c\rangle$ correspond to the matrix U_{P_q} and can be computed in polynomial time of n.

Example. Let H_q be a q-dimensional Hadamard matrix$\left(q = 2^n\right)$, that is, when $H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ then $H_q = H_2 \otimes H_{q/2} = \begin{pmatrix} H_{q/2} & H_{q/2} \\ H_{q/2} & -H_{q/2} \end{pmatrix}$. The QTM can execute the matrix $\frac{1}{\sqrt{q}} H_q$ in $O(n)$ time. Further, when let H_{kj} be the k-th row and j-th column element of H_q then $H_{kj} = W_{br(k)}\left(\frac{j}{q}\right)$ where for $k = \sum_{i=0}^{n-1} k_i 2^i \left(k_i \in \{0,1\}\right), br(k) = \sum_{i=0}^{n-1} k_{n-i-1} 2^i$ (for a bit string, its reverse order is obtained). The QTM can also execute this procedure in $O(n)$ time. Consequently, the QTM can execute the transform P_q in $O(n)$ time.

Relationships among some Walsh transforms

For the value $a = \sum_{i=0}^{n-1} a_i 2^i$, $a_i \in \{0,1\}$, let

$$g_i\left(i = 0,1,\ldots,n-1\right) = \begin{cases} g_{n-1} = a_{n-1} \\ g_i = a_{i+1} \oplus a_i \left(0 \le i \le n-2\right) \end{cases}.$$

Gray code $g(a)$ of a is defined by $g(a) = \sum_{i=0}^{n-1} g_i 2^i$. Now, when $W\left(a, \frac{b}{q}\right)$ and $H_a\left(\frac{b}{q}\right)$ are discrete *Walsh-Paley* function and *Walsh-Hadamard* function, respectively, the relationships among Walsh functions are $W\left(a, \frac{b}{q}\right) = W_{g(a)}\left(\frac{b}{q}\right)$ and $H_a\left(\frac{b}{q}\right) = W_{br(a)}\left(\frac{b}{q}\right)$. Moreover, we can describe them by

$W\left(a, \frac{b}{q}\right) = (-1)^{br(g(a))\cdot b}$	$W_a\left(\frac{b}{q}\right) = (-1)^{br(a)\cdot b}$	$H_a\left(\frac{b}{q}\right) = (-1)^{a\cdot b}$

Simon's problem and algorithm

Simon (1994, 1997) was the first to show a nice and simple problem with expected polynomial time QA but with no polynomial time randomized algorithm.

The qualitative description of Simon's problem

A function $f : \{0,1\}^n \to \{0,1\}^n$ is given as an oracle, with the promise that there exists an $s \in \{0,1\}^n$ (known as the "hidden secret") such that $f(x) = f(y)$ iff $x \oplus y = s$. Notice that if $s = 0^n$, then f is a permutation, and otherwise f is two-to-one function. The problem is to tell if $s = 0^n$.

Simon's algorithm works as follows. One starts with $2n$ qubits, separated into two n-qubit registers. Originally one initializes the states to $|\phi_0\rangle = |0^n\rangle|0^n\rangle$. Next, one applies the Hadamard operator to the first register and then the oracle operator $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$. The state becomes $|\phi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle|f(x)\rangle$.

Next, the second register is measured and discarded:

- If $s = 0^n$, then the measurement result is $|\phi_2\rangle = |x\rangle$ for a random $x \in \{0,1\}^n$;
- If $s \neq 0^n$, then the measurement is $|\phi_2'\rangle = \frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)$ for a random x

Next, a Hadamard operator is applied to the first register.

In the case $s = 0^n$, the result is $|\phi_3\rangle = |y\rangle$ for a random y; in the case $s \neq 0^n$, the result $|\phi_3'\rangle = |y\rangle$ for a random y such that $y \cdot s = 0$.

Finally, one measurement the first register and obtain y.

Repeating the experiment $O(n)$ times, one can solve for s by using Gaussian elimination and distinguish the case $s = 0^n$ from the case $s \neq 0^n$.

Mathematical model of Simon's problem: Simon's XOR Problem

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function such that either f is one-to-one and there exists a single non-zero $s \in \{0,1\}^n$ such that $\forall x \neq x' \left(f(x) = f(x') \Leftrightarrow x' = x \oplus s \right)$. The task is to determine of the above conditions for f and, in the second case to determine also s.

To solve the problem two registers are used, both with n qubits and the initial states $|0^n\rangle$, and (expected) $O(n)$ repetitions of the following version of the Hadamard-twice scheme:

Step	Computational algorithm
1	Apply the Hadamard transformation on the first register, with the initial value $\|0^n\rangle$, to produce the superposition $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, 0^n\rangle$
2	Apply U_f to compute $\|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, f(x)\rangle$
3	Apply Hadamard transformation on the first register to get $\frac{1}{2^n} \sum_{x,y \in \{0,1\}^n} (-1)^{x \cdot y} |y, f(x)\rangle$
4	Observe the resulting state to get a pair $(y, f(x))$

Case 1: f *is one-to-one*. After performing the first three steps of the above procedure all possible states $|y, f(x)\rangle$ in the superposition are distinct and the absolute value of their amplitudes is the same, namely

$\frac{1}{2^n}$. Therefore, $n-1$ independent applications of the scheme Hadamard-twice produce $n-1$ pairs $\{y_1, f(x_1)\},\ldots,\{y_{n-1}, f(x_{n-1})\}$, distributed uniformly and independently over all pairs $\{y, f(x)\}$.

Case 2: *There is some* $s \neq 0^n$ *such that* $\forall x \neq x' \left(f(x) = f(x') \Leftrightarrow x' = x \oplus s \right)$. In such a case for each y and x the states $|y, f(x)\rangle$ and $|y, f(x \oplus s)\rangle$ are identical. Their total amplitude $\alpha(x, y)$ has the value $\alpha(x, y) = \frac{1}{2^n}\left[(-1)^{x \cdot y} + (-1)^{(x \oplus) \cdot y}\right]$. If $y \cdot s \equiv 0 \bmod 2$, then $x \cdot y \equiv (x \oplus s) \cdot y \bmod 2$ and therefore $\alpha(x, y) = 2^{-n+1}$; otherwise $\alpha(x, y) = 0$. Therefore, n independent applications of the scheme Hadamard-twice yield $n-1$ independent pairs $\{y_1, f(x_1)\},\ldots,\{y_{n-1}, f(x_{n-1})\}$ such that $y_i \cdot s \equiv 0 \bmod 2$ for all $1 \leq i \leq n-1$.

Remark. In both cases, after $n-1$ repetitions of the scheme Hadamard-twice, $n-1$ vectors y_i, $1 \leq i \leq n-1$, are obtained. If these vectors are linearly independent, then the system of $n-1$ linear equations in $\mathbb{Z}_2$, $y_i \cdot s = 0$ can be solved to obtain s. In Case 2, if f is two-to-one, s obtained in such a way is the one to be found. In Case 1, s obtained in such a way is a random string. To distinguish these two cases, it is enough to compute $f(0)$ and $f(s)$. If $f(0) \neq f(s)$, then f is one-to-one. If the vector obtained by the scheme Hadamard-twice are not linearly independent, then the whole process has to be repeated.

As shown in the next lemma, the vectors y_i, $1 \leq i \leq n-1$, obtained in this way are linearly independent with probability at least $\frac{1}{4}$. The total expected computation time is therefore $O(nt(n) + g(n))$, where $t(n)$ is time needed to compute f on inputs of length n and $g(n)$ is time needed to solve the system of n linear equations in $\mathbb{Z}_2$.

Lemma: If u is a non-zero binary vector of length n, then $n-1$ randomly chosen binary vectors of length n such that $u \cdot y \equiv 0 \bmod 2$ are linearly independent with probability at least $\frac{1}{4}$.

Let us consider the relationship between the structure of Simon's algorithm and Walsh transforms.

Walsh transforms and Simon's Problem

First, let us consider a function $W : \mathbb{N} \times \mathbb{N} \times \left\{2^n \middle| n \in \mathbb{N}^+\right\} \to \{-1,1\}$. This function satisfies the above mentioned conditions. For this case we consider solving Simon's problem by using the transform W_q.

Simon's Problem. Let f be a function $f : \{0,1\}^n \to \{0,1\}^m \ (m \geq n)$ such that (A) the function f is a one-to-one function, or (B) there exists a non-trivial s satisfying

$$\forall x \neq x' \left[f(x) = f(x') \Leftrightarrow x' = x \oplus s \right].$$

Then, the problem is to decide which condition the function f satisfies. Moreover, in the case of (B), find the value of s.

Algorithm solution. Let $q = 2^n$. In order to solve Simon's problem, first, a QTM computes all of the values of f in quantum parallel computation:

$$|0\rangle|0\rangle \to \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|0\rangle \to \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|f(a)\rangle.$$

When let $|a\rangle|b\rangle$, call $|a\rangle$ (respectively, $|b\rangle$) the first register (respectively the second register).

Remark. The original computation for the first register is the Fourier transform. However, the QTM can obtain the same result by using the transform W_q.

For the first register, the QTM executes the transform W_q:

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle|f(a)\rangle \xrightarrow{W_q} \frac{1}{q}\sum_{a=0}^{q-1}\left(\sum_{c=0}^{q-1}W(a,c,q)|c\rangle\right)|f(a)\rangle.$$

If the function f satisfies the condition (B) [i.e., $f(a)=f(a\oplus s)$], for each c, two configurations $|c\rangle f(a)$ and $|c\rangle f(a\oplus s)$ are equal and the probability amplitude $\alpha(a,c)$ corresponding to them is

$$\alpha(a,c)=\frac{1}{q}\{W(a,c,q)+W(a\oplus s,c,q)\}=\frac{1}{q}W(a,c,q)\{1+W(s,c,q)\}.$$

Therefore, the configuration corresponding to $W(s,c,q)=-1$ is erased. Then, for the given function $W(s,c,q)$, the QTM can obtain c satisfying $W(s,c,q)=1$. Moreover, when the QTM repeats procedure above several times, it can obtain some different values of c (although whether or not we can find s depends on the structure of the function W). On the other hand, when the function f satisfies condition (A), the QTM can decide whether the function f satisfies either (A) or (B).

Remark. Moreover, if the QTM can execute both the computation of the transform W_q and the computation of the function f in polynomial time of $\log_2 q$, it can execute the total procedure in polynomial time of $\log_2 q$. The algorithm above becomes Simon's algorithm: when $W(s,c,2^n)=1, s\cdot c=0$.

We describe a problem that is efficiently solved using one of the algorithms mentioned above.

Generalized QA based on using of Walsh transform

Let us consider the following problem:

Let $f:\{0,1\}^n\to\{0,1\}^m\ (m\geq n)$ be a function such that there exists a nontrivial s satisfying

$$\forall x\neq x'\left(f(x)=f(x')\Leftrightarrow x'=x\oplus s\right).$$

Then, the problem is to find a nontrivial b satisfying $s\cdot b=0$.

Remark. This problem is an extended version of the Simon's problem, that is, in the current *Problem*, the range of the function f is extended from $\{0,1\}^{n-1}$ to $\{0,1\}^m\ (m\geq n)$. Simon shows that any PTM needs an exponential time of n to solve it. In the following, we describe a QA to efficiently solve this Problem.

Solution of the Problem. We define a Walsh-Hadamard transform H_q by

$$|a\rangle \xrightarrow{H_q} \frac{1}{\sqrt{q}}\sum_{b=0}^{q-1}H_a\left(\frac{b}{q}\right)|b\rangle.$$

Moreover, for a given $g\in\{0,1\}^n$, we define two functions F_1 and F_2:

$$F_1(a)=\begin{cases}f(a), \textit{if } f(a)>f(a\oplus g)\\ f(a\oplus g), \quad \textit{otherwise}\end{cases} \qquad F_2(a)=\begin{cases}0, \textit{if } f(a)>f(a\oplus g)\\ 1, \quad \textit{otherwise}\end{cases}.$$

We then define U_{F_1} and G_{F_2} by $|a\rangle|0\rangle \xrightarrow{U_{F_1}} |a\rangle|F_1\rangle$, and $|a\rangle \xrightarrow{G_{F_2}} (-1)^{F_2(a)}|a\rangle$. The function F_1 can be computed in the following way: First, the QTM computes $f(a)-f(a\oplus g)$. The QTM takes the value of $f(a)$ if the value of $f(a)-f(a\oplus g)$ is plus, otherwise it takes the value of $f(a\oplus g)$. Therefore, the main procedure computing the function F_1 is to compute a linear function whose inputs are values of f.

Remark. We can construct transforms computing linear functions (in general, polynomial functions). Moreover, since the number of inputs in the linear function is constant, the complexity $T_f(n)$ is time complexity of f (we can evaluate the time complexity of F_2 in a similar way).

Algorithm of solution

First, for an a, the QTM selects a $g(\neq 0)$, such that $f(a)\neq f(a\oplus g)$, and computes the function F_1 [if $f(a)=f(a\oplus g)$, then $s=g$, and we can easily find a nontrivial b such that $s\cdot b=0$]: $|0\rangle|0\rangle \xrightarrow{H_{2^n}} \frac{1}{\sqrt{2^n}}\sum_{a=0}^{2^n-1}|a\rangle|0\rangle \xrightarrow{U_{F_1}} \frac{1}{\sqrt{2^n}}\sum_{a=0}^{2^n-1}|a\rangle|F_1(a)\rangle$.

Next, the QTM observes the second register (thus, in the following we omit the second register). By $F_1(a)=F_1(a\oplus g)$, and $\frac{1}{2}\left[|a\rangle+|a\oplus s\rangle+|a\oplus g\rangle+|a\oplus g\oplus s\rangle\right]$. Moreover, the QTM executes the transform G_{F_2}. By $(-1)^{F_2(a\oplus g)}=-(-1)^{F_2(a)}$,

$\frac{1}{2}\left[\lvert a\rangle+\lvert a\oplus s\rangle+\lvert a\oplus g\rangle+\lvert a\oplus g\oplus s\rangle\right]$	$\xrightarrow{G_{F_2}}$	$\frac{1}{2}\begin{bmatrix}(-1)^{F_2(a)}\lvert a\rangle+(-1)^{F_2(a\oplus s)}\lvert a\oplus s\rangle \\ +(-1)^{F_2(a\oplus g)}\lvert a\oplus g\rangle+(-1)^{F_2(a\oplus g\oplus s)}\lvert a\oplus g\oplus s\rangle\end{bmatrix}$
	$=$	$\frac{1}{2}(-1)^{F_2(a)}\left[\lvert a\rangle+\lvert a\oplus s\rangle-\lvert a\oplus g\rangle-\lvert a\oplus g\oplus s\rangle\right]$

Finally, the QTM executes the Walsh-Hadamard transform H_{2^n}:

$\frac{1}{2}(-1)^{F_2(a)}\begin{bmatrix}\lvert a\rangle+\lvert a\oplus s\rangle \\ -\lvert a\oplus g\rangle-\lvert a\oplus g\oplus s\rangle\end{bmatrix}$	$\xrightarrow{H_{2^n}}$	$\frac{1}{2\sqrt{2^n}}(-1)^{F_2(a)}\left[\sum_{b=0}^{2^n-1}\begin{bmatrix}H_a\left(\frac{b}{2^n}\right)+H_{a\oplus s}\left(\frac{b}{2^n}\right) \\ -H_{a\oplus g}\left(\frac{b}{2^n}\right) \\ -H_{a\oplus g\oplus s}\left(\frac{b}{2^n}\right)\end{bmatrix}\lvert b\rangle\right]$
	$=$	$\frac{1}{2\sqrt{2^n}}(-1)^{F_2(a)}\sum_{b=0}^{2^n-1}\left[\begin{matrix}H_a\left(\frac{b}{2^n}\right)\left(1+H_s\left(\frac{b}{2^n}\right)\right) \\ \times\left(1-H_g\left(\frac{b}{2^n}\right)\right)\end{matrix}\right]\lvert b\rangle$

Then, we can find b satisfying $H_s\left(\frac{b}{2^n}\right)=1$ and $H_g\left(\frac{b}{2^n}\right)=-1$, that is, $s\cdot b=0$ and $g\cdot b=1$, in polynomial time of n and $T_f(n)$.

Remark. Since any Walsh function satisfies the conditions of $W(a,b,q)$, the QTM can solve Simon's problem in polynomial time of n by using any Walsh function. Furthermore, if we suppose that all of the transforms corresponding to the Walsh functions can be executed in the same time (i.e., if we suppose that we can construct such quantum networks executable in the same time), we can obtain the required code most efficiently by using the function corresponding to the code. For example, when we use the function $W\left(a,\frac{b}{q}\right)$, we can obtain $br(g(s))$ instead of s because we obtain c satisfying $br(g(s))\cdot c=0$.

Deutsch-Jozsa QA and Walsh-Hadamard transformation

Deutsch and Jozsa suggested that there exists a problem that a QTM may solve it exponentially faster than any DTM. Let us briefly repeat the discussion of this problem.

Deutsch-Jozsa problem.

Let us consider a function $f:\mathbb{Z}_{2N}\to\{0,1\}\ (N\in\mathbb{Z}^+)$ such that (A) the function is not constant (at 0 or 1), or (B) the sequence of all of the values of f, $f(0), f(1),\ldots, f(2N-1)$ does not contain exactly N zeros. Then the problem is to decided which condition the function f satisfies.

First, we briefly describe an algorithm (Deutsch-Jozsa QA) to solve this problem on a QTM. Let S_2 be a unitary matrix (a unitary transform) $S_2=\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$. For one qubit, this matrix operates and transforms as follows: $|a\rangle\xrightarrow{S_2}(-1)^a|a\rangle$. Moreover, the function f is computed as

$$|a\rangle|b\rangle\xrightarrow{U_f}|a\rangle|b\oplus f(a)\rangle.$$

We denote the initial configuration by a pair of an input (the first register) and an output register (the second register), $|0\rangle|0\rangle$. Then the QTM executes the following computations.

$\|0\rangle\|0\rangle$	$\xrightarrow{W_{2N}}$	$\frac{1}{\sqrt{2N}}\sum_{a=0}^{2N-1}\|a\rangle\|0\rangle$
action on the second register	$\xrightarrow{U_f}$	$\frac{1}{\sqrt{2N}}\sum_{a=0}^{2N-1}\|a\rangle\|f(a)\rangle$
action on the first register	$\xrightarrow{S_2}$	$\frac{1}{\sqrt{2N}}\sum_{a=0}^{2N-1}(-1)^{f(a)}\|a\rangle\|f(a)\rangle$
action on the second register	$\xrightarrow{U_f}$	$\frac{1}{\sqrt{2N}}\sum_{a=0}^{2N-1}(-1)^{f(a)}\|a\rangle\|0\rangle$

For the final superposition of configurations (let $|\psi\rangle$ be the superposition), the probability $P(\lambda_\varphi)$ of observing the eigenvalue λ_φ that corresponds to $|\varphi\rangle = \frac{1}{\sqrt{2N}} \sum_{a=0}^{2N-1} |a\rangle|0\rangle$ is

$$P(\lambda_\varphi) = |\langle\varphi|\psi\rangle|^2 = \left| \frac{1}{2N} \sum_{a=0}^{2N-1} (-1)^{f(a)} \right|^2 .$$

Therefore the probability of solution observable as $P(\lambda_\varphi) = 1$ if f is a constant function, and, $P(\lambda_\varphi) = 0$, if exactly half the values of f are 0. Consequently, when we take the contrapositive, condition (A) is true if λ_φ is not obtained, and otherwise condition (B) is true.

Remark. Thus, Deutsch-Jozsa algorithm gives a method of efficiently solving problems on QTMs by using observations effectively. On the other hand, as mentioned, Simon's algorithm is a method of retaining only the required configurations by using *interference* a superposition of configurations is solved. Moreover, by a Fourier transform, Shor developed algorithms to factor integers and to find discrete logarithms efficiently on QTMs by using interference (inspired by Simon's algorithm). We show that Deutsch-Jozsa's problem can also be efficient solved by using the Walsh-Paley transform P_q and by using *interference* (we can also show that it can be solved by other Walsh transforms).

Example. For simplicity, let $2N = 2^n$. A QTM executes the same computations as above until the *final superposition* $|\psi\rangle$ of configurations. For the vector $|\psi\rangle$, the QTM executes the Walsh-Paley transform:

$\|\psi\rangle$	=	$\frac{1}{\sqrt{2^n}} \sum_{a=0}^{2^n-1} (-1)^{f(a)} \|a\rangle\|0\rangle$
action on the first register	$\xrightarrow{P_{2^n}}$	$\frac{1}{2^n} \sum_{a=0}^{2^n-1} \sum_{c=0}^{2^n-1} W_a\left(\frac{c}{2^n}\right) (-1)^{f(a)} \|c\rangle\|0\rangle$
result	=	$\sum_{c=0}^{2^n-1} \alpha_c \|c\rangle\|0\rangle$
α_c	=	$\frac{1}{2^n} \sum_{c=0}^{2^n-1} W_a\left(\frac{c}{2^n}\right) (-1)^{f(a)}$

where α_c is a value depending on the function f. Moreover, all of the values of the function $W_a\left(\frac{c}{2^n}\right)$ are 1 if $c = 0$, otherwise exactly half the values of the function are 1 (and half the remaining values of it are -1). Then, if f is a constant function, $\alpha_c = \pm\delta_{c0}$ and the configuration after the computation becomes $\pm|0\rangle|0\rangle$. On the hand, if exactly half the values of f are 0, $\alpha_0 = 0$ and the configuration after the computation becomes $\sum_{c=1}^{2^n-1} \alpha_c |c\rangle|0\rangle$. Here, when we observe the first register, the probability of observing $|0\rangle$ (more precisely, the probability of observing the eigenvalues corresponding to it) is 1 if f is a constant function, and the probability of observing $|0\rangle$ is 0 if exactly half the values of f are 0. Consequently, under the same evaluation as above, when we take the contrapositive, condition (A) is true if $|0\rangle$ is not obtained, and otherwise condition (B) is true.

Remark. Deutsch-Jozsa's problem is a type of decision-making problem of choosing between two things. Finally, we modify the problem to one that includes forms of search problems and derive more general algorithms to solve them.

Example. Let f be a function such that $f:\{0,1\}^n \to \{0,1\}^m \ (m \geq n)$. We denote two unitary transforms U_α and U_β by

$\lvert a\rangle$	$\xrightarrow{U_\alpha}$	$\sum_{b=0}^{2^n-1}\alpha_{ab}\lvert b\rangle \left(\sum_{b=0}^{2^n-1}\alpha_{ab}^*\alpha_{bc}=\delta_{ac}\right)$	$\lvert\beta\rangle$	$\xrightarrow{U_\beta}$	$\sum_{b=0}^{2^n-1}\beta_{ab}\lvert b\rangle \left(\sum_{b=0}^{2^n-1}\beta_{ab}^*\beta_{bc}=\delta_{ac}\right)$

Moreover, we define a unitary transform G by $\lvert a\rangle \xrightarrow{G} g(a)\lvert a\rangle$ where $g(a)$ is a function satisfying $\lvert g(a)\rvert^2 = 1$ under some conditions described later (the function S_2 is an instance of G).

Algorithm. First, for an initial configuration $\lvert 0\rangle\lvert 0\rangle$ of a QTM (although we do not necessarily select the initial configuration as $\lvert 0\rangle\lvert 0\rangle$), we form a superposition of all of the input values of the function f by using the transform U_β: $\lvert 0\rangle\lvert 0\rangle \xrightarrow{U_\beta} \sum_{a=0}^{2^n-1}\beta_{0a}\lvert a\rangle\lvert 0\rangle$. Next, the QTM executes the transforms $U_f, G,$ and U_f, as following:

$\sum_{a=0}^{2^n-1}\beta_{0a}\lvert a\rangle\lvert 0\rangle$	$\xrightarrow{U_f}$	$\sum_{a=0}^{2^n-1}\beta_{0a}\lvert a\rangle\lvert f(a)\rangle$
result of $G-$*action*	$\xrightarrow{G}$	$\sum_{a=0}^{2^n-1}\beta_{0a}g(f(a))\lvert a\rangle\lvert f(a)\rangle$
action on the second register	$\xrightarrow{U_f}$	$\sum_{a=0}^{2^n-1}\beta_{0a}g(f(a))\lvert a\rangle\lvert 0\rangle$

Finally, the QTM executes the transform U_α:

$$\sum_{a=0}^{2^n-1}\beta_{0a}g(f(a))\lvert a\rangle\lvert 0\rangle \xrightarrow{U_\alpha} \sum_{a=0}^{2^n-1}\sum_{c=0}^{2^n-1}\beta_{0a}g(f(a))\alpha_{ac}\lvert c\rangle\lvert 0\rangle.$$

If we suppose that $\beta_{0a}g(f(a)) = p\alpha_{ba}^*$ (where, $\lvert p\rvert^2 = 1$) the superposition of the configurations becomes

$\sum_{a=0}^{2^n-1}\sum_{c=0}^{2^n-1}\beta_{0a}g(f(a))\alpha_{ac}\lvert c\rangle\lvert 0\rangle$	$=$	$\sum_{a=0}^{2^n-1}\sum_{c=0}^{2^n-1}p\alpha_{ba}^*\alpha_{ac}\lvert c\rangle\lvert 0\rangle$
	$=$	$\sum_{c=0}^{2^n-1}p\delta_{bc}\lvert c\rangle\lvert 0\rangle$
	$=$	$p\lvert b\rangle\lvert 0\rangle$

and we can obtain a value b uniquely.

Remark. For a given f, the relationship between the function f and the value b is decided; however, as in the example mentioned below (Deutsch-Jozsa-type search problem), we can also assign the value b to each function as identification number.

Next, let us consider more general algorithm.

Example. Suppose that $\beta_{0a} g\left(f\left(a\right)\right) = \sum_{b \in S_b} p_b \alpha_{ba}^*$ (where $\sum_{b \in S_b} \left|p_b\right|^2 = 1$) where for any two different elements $b_1, b_2 \in \mathbb{Z}_k$, the set S_b ($b \in \mathbb{Z}_k, k \in \mathbb{N}$) satisfies $S_{b_1} \cap S_{b_2} = \varnothing$ and $\bigcup_{b=0}^{k} S_b = \mathbb{Z}_{2^n}$ (consider that each condition of solving a problem corresponds to a number, i.e., an element of the set). Then, the superposition of configurations after the computation above becomes

$\sum_{a=0}^{2^n-1}\sum_{c=0}^{2^n-1} \beta_{0a} g\left(f\left(a\right)\right)\alpha_{ac}\left\|c\right\rangle\left\|0\right\rangle$	=	$\sum_{a=0}^{2^n-1}\sum_{c=0}^{2^n-1}\sum_{b \in S_b} p_b \alpha_{ba}^* \alpha_{ac}\left\|c\right\rangle\left\|0\right\rangle$
	=	$\sum_{b \in S_b} p_b \sum_{c=0}^{2^n-1} \delta_{bc}\left\|c\right\rangle\left\|0\right\rangle$
	=	$\sum_{b \in S_b} p_b \left\|b\right\rangle\left\|0\right\rangle$

and we can uniquely decide to which correction of properties the given function f belongs.

Example. For an error-bounded algorithm, we suppose that

$$\beta_{0a} g\left(f\left(a\right)\right) = \sum_{b \in S_b} p_b \alpha_{ba}^* + \sum_{b' \in \mathbb{Z}_{2^n} - S_b} p_{b'} \alpha_{b'a}^*, \quad \sum_{b \in S_b} \left|p_b\right|^2 + \sum_{b' \in \mathbb{Z}_{2^n} - S_b} \left|p_{b'}\right|^2 = 1, \sum_{b \in S_b} \left|p_b\right|^2 \geq \frac{2}{3}.$$

(for $b_1, b_2 \in \mathbb{Z}_{2^n}$, we may select $S_{b_1} \cap S_{b_2} \neq \varnothing$). Then, the superposition of configurations after the computation above becomes

$\sum_{a=0}^{2^n-1}\sum_{c=0}^{2^n-1} \beta_{0a} g\left(f\left(a\right)\right)\alpha_{ac}\left\|c\right\rangle\left\|0\right\rangle$	=	$\sum_{a=0}^{2^n-1}\sum_{c=0}^{2^n-1}\sum_{b \in S_b} p_b \alpha_{ba}^* \alpha_{ac}\left\|c\right\rangle\left\|0\right\rangle + \sum_{a=0}^{2^n-1}\sum_{c=0}^{2^n-1}\sum_{b' \in \mathbb{Z}_{2^n} - S_b} p_{b'} \alpha_{b'a}^* \alpha_{ac}\left\|c\right\rangle\left\|0\right\rangle$
	=	$\sum_{b \in S_b} p_b \left\|b\right\rangle\left\|0\right\rangle + \sum_{b' \in \mathbb{Z}_{2^n} - S_b} p_{b'} \left\|b'\right\rangle\left\|0\right\rangle$

and we can obtain an element in S_b with probability more than $\frac{2}{3}$.

Remark. The algorithms above reset the output value (the second register) to 0 in the course of computation; however, we can easily modify these algorithms to algorithms with classification by the output values, like Simon's and Shor's algorithms.

Now, we show that even if we modify the Deutsch-Jozsa problem (decision-making problem) to the form of search problems, there also exists a problem that it solved efficiently.

Example: *Deutsch-Jozsa search problem*. Let $\frac{1}{\sqrt{2^n}} p_{ab} \left(a, b \in \mathbb{Z}_{2^n}\right)$ be the a-th row and b-th column element of a matrix corresponding to the Walsh-Paley transform. Moreover, we define 2^n functions $f_b\left(a\right) = \frac{p_{ab} + 1}{2}$ [more precisely, let $f_b\left(a\right)$ be a function outputting such a value].

Then, for a given function f among 2^n functions $f_b\left(a\right)$, the problem is to decide to which function the given function f belongs, i.e., find b. In order to solve this problem, we investigate the general algorithms above.

Thus, when $\alpha_{ab} = \beta_{ab} \frac{1}{\sqrt{2^n}} W_a\left(\frac{b}{2}\right)$ and $g\left(f\left(a\right)\right) = \left(-1\right)^{f(a)}$, then

$$\beta_{0a} g\left(f(a)\right) = \frac{1}{\sqrt{2^n}} W_0\left(\frac{a}{2^n}\right)(-1)^{f(a)} = \frac{1}{\sqrt{2^n}}(-1)^{\frac{p_{ab}+1}{2}} = -\alpha_{ba}^*$$

and it satisfies the condition above.

Then, the final superposition of configurations is $-|b\rangle|0\rangle$ and we can obtain the value b uniquely.

References

1. Gruska J. Quantum computing. – Advanced Topics in Computer Science Series, McGraw-Hill Companies, London. – 1999.
2. Nielsen M.A. and Chuang I.L. Quantum computation and quantum information. – Cambridge University Press, Cambridge, Englandю – 2000.
3. Hirvensalo M. Quantum computing. – Natural Computing Series, Springer-Verlag, Berlinю – 2001.
4. Hardy Y. and Steeb W.-H. Classical and quantum computing with C++ and Java Simulations. – Birkhauser Verlag, Basel. – 2001.
5. Hirota O. The foundation of quantum information science: Approach to quantum computer (in Japanese). – Japan. – 2002.
6. Pittenberg A.O. An introduction to quantum computing and algorithms. – Progress in Computer Sciences and Applied Logic. – Vol. 19. – Birkhauser. – 1999.
7. Brylinski F.K. and Chen G. (Eds). Mathematics of quantum computation. – Computational Mathematics Series. – CRC Press Co. – 2002.
8. Wang X. Krawtchouk matrices, Krawtchouk polynomials and trace formula. - Southern Illinois University Carbondale, 2008.
9. Chami P.S., Sing B. and Sookoo N. Generalized Krawtchouk polynomials using Hadamard matrices // arXiv:1307.5777v1 [math.CO] 22 Jul 2013.
10. Feinsilver P. and Kocik J. Krawtchouk matrices from classical and quantum random walks // arXiv:quant-ph/0702173v1 16 Feb 2007.
11. Feinsilver P. and Kocik J. Krawtchouk polynomials and Krawtchouk matrices // arXiv:quant-ph/0702073v1 7 Feb 2007.

Appendix 1: Tensor products

Let V be a d-dimensional vector space over $\boldsymbol{R}$. We fix an orthonormal basis $\{e_0,\ldots,e_\delta\}$ with $d = 1+\delta$. Denote tensor powers of V by $V^{\otimes N}$, so that $V^{\otimes 2} = V \otimes V$, etc. A basis for $V^{\otimes N}$ is given by all N -fold tensor products of the basis vectors e_i,

$$|n_1 n_2 \ldots n_N\rangle = e_{n_1} \otimes e_{n_2} \otimes \cdots \otimes e_{n_N}$$

Note that we can label these d^N basis elements by all numbers 0 to d^N -1 and recover the tensor products by expressing these numbers in base d, putting leading zeros so that all extended labels are of length N.

Now let $\{A_i : 1 \le i \le N\}$ be a set of N linear operators on V. On $V^{\otimes N}$, the linear operator $A = A_1 \otimes A_2 \otimes \cdots \otimes A_N$ acts on a basis vector $|n_1 n_2 \ldots n_N\rangle$ by

$$A|n_1 n_2 \ldots n_N\rangle = A_1 e_{n_1} \otimes \cdots \otimes A_N e_{n_N}$$

This needs to be expanded and terms regrouped using bilinearity.

If A and B are two $d \times d$ matrices, the matrix corresponding to the operator $A \otimes B$ is the Kronecker product, $d^2 \times d^2$ matrix having the block form:

$$\begin{pmatrix} a_{00}B & \cdots & a_{0\delta}B \\ a_{10}B & \cdots & a_{1\delta}B \\ \vdots & \vdots & \vdots \\ a_{\delta 0}B & \cdots & a_{\delta\delta}B \end{pmatrix}$$

associating, from the left for higher-order tensor products. The rows and columns of the matrix of a linear operator acting on $V^{\otimes N}$ are conveniently labelled by associating to each basic tensor $|n_1 n_2 \ldots n_N\rangle$ the corresponding integer label $\sum_{k=1}^{N} n_k d^{N-k}$, which thus provides a canonical ordering.

Appendix 2: Symmetric tensor spaces

Here we review symmetric tensor spaces as spaces of polynomials in commuting variables.

The space $V^{\otimes N}$ can be mapped onto the space of symmetric tensors, $V^{\otimes_S N}$ by identifying basis vectors (in $V^{\otimes N}$) that are equivalent under all permutations. Alternatively, one can identify the basic tensor $|n_1 n_2 \ldots n_N\rangle$ with the monomial $x_{n_1} x_{n_2} \cdots x_{n_N}$ in the commuting variables $x_0, \ldots, x_\delta$. Hence we have a linear map from tensor space into the space of polynomials, itself isomorphic to the space of symmetric tensors: $\bigcup_{N \geq 0} V^{\otimes N} \to R[x_0, \ldots, x_\delta] \cong \bigcup_{N \geq 0} V^{\otimes_S N}$. In the symmetric tensor space, tensor labels need to count only 'occupancy', that is, the number of times a basis vector of V occurs in a given basic tensor of $V^{\otimes N}$. We indicate occupancy by a multi-index which is the exponent of the corresponding monomial. The dimension of $V^{\otimes_S N}$ is thus $\dim V^{\otimes_S N} = \begin{pmatrix} N+d-1 \\ d-1 \end{pmatrix}$ that is, the number of monomials homogeneous of degree N.

Given an operator A on V, let $A_N = A^{\otimes N}$. Then A_N induces an operator $\overline{A}_N$ on $V^{\otimes_S N}$ from the action of A on polynomials, which we call the *symmetric representation of A in degree N*. For convenience we work dually with the tensor components rather with the action on the basis vectors. Denote the matrix elements of the action of $\overline{A}_N$ by $\overline{A}_{mn}$. If A has matrix entries A_{ij} let

$$y_i = \sum_j A_{ij} x^j .$$

Then the matrix elements of the symmetric representation are defined by the relation (expansion):

$$y_0^{m_0} \cdots y_\delta^{m_\delta} = \sum_n \overline{A}_{mn} x_0^{n_0} \cdots x_\delta^{n_\delta}$$

with multi-indices m and n.

Successive application of A_1 then A_2 shows that mapping to the symmetric representation is an algebra homomorphism, i.e.,

$$\overline{A_1 A_2} = \overline{A}_1 \overline{A}_2 .$$

Explicitly, in basis notation

$$\overline{(A_1 A_2)}_{mn} = \sum_r \left(\overline{A}_1\right)_{mr} \left(\overline{A}_2\right)_{rn} .$$

Define the symmetric trace in degree N of A as the trace of the matrix elements of $\overline{A}_N$, i.e., the sum of the diagonal matrix elements:

$$\mathrm{tr}_{Sym}^{N} A = \sum_{|m|=N} \bar{A}_{mm}$$

with $|m|$ denoting, as usual, the sum of the components of m. Observe that if A is upper-triangular, with eigenvalues $\lambda_1,\ldots,\lambda_d$, then the trace of this action on the space of polynomials homogeneous of degree N is exactly $h_N(\lambda_1,\ldots,\lambda_d)$, the N th homogeneous symmetric function in the λ 's.

We recall a useful theorem on calculating the symmetric trace. Since the mapping from A to $\bar{A}_N$ is a homomorphism, a similarity transformation on A extends to one on $\bar{A}_N$ thus preserving traces. Now, any matrix is similar to an upper-triangular one with the same eigenvalues, thus follows

Theorem: Symmetric trace theorem *Denoting by* tr_{Sym}^{N} *the trace of the symmetric representation on polynomials homogeneous of degree* N ,

$$\frac{1}{\det(I-tA)} = \sum_{N=0}^{\infty} t^N \mathrm{tr}_{Sym}^{N} A$$

Proof: With $\{\lambda_i\}$ denoting the eigenvalues of A,

$$\frac{1}{\det(I-tA)} = \prod_i \frac{1}{1-t\lambda_i}$$

$$= \sum_{N=0}^{\infty} t^N h_N(\lambda_1,\ldots,\lambda_2)$$

$$= \sum_{N=0}^{\infty} t^N \mathrm{tr}_{Sym}^{N} A$$

as stated above.

Remark Note that this result is equivalent to MacMahon's Master Theorem in combinatorics.

Appendix 3: General properties of Walsh-Hadamard transformation $W(a,b,q)$

Let us consider a function $W: \mathcal{N}\times\mathcal{N}\times\{2^n | n\in\mathbb{Z}^+\}\to\{-1,1\}$. The function satisfies the following conditions:

$W(0,b,q)=1$	$W(a\oplus c,b,q)=W(a,b,q)W(c,b,q)$
$W(a,b,q)=W(b,a,q)$	$\sum_{c=o}^{q-1} W(a,c,q)W(c,b,q)=q\delta_{ab}$

where the operator $\oplus$ is a bitwise exclusive-OR (XOR). Moreover, we define the transform W_q as follows: $|a\rangle \xrightarrow{W_q} \frac{1}{\sqrt{q}}\sum_{c=0}^{q-1} W(a,c,q)|c\rangle$. When the a-th row and b-th column element of a matrix U_{W_q} is $\frac{1}{\sqrt{q}}W(a,b,q)$ (unless otherwise specified, the rows and columns will be indexed beginning with 0), the matrix is a unitary matrix because of the definition of $W(a,b,q)$. Thus, the Quantum Turing Machine (QTM) can execute the transform W_q.

Example. Let $W\left(a,b,q=2^n\right)=\left(-1\right)^{a\cdot b}$ where $a\cdot b$ is an inner product of a and b, i.e., for $a=\sum_{i=0}^{n-1}a_i 2^i$ and $b=\sum_{i=0}^{n-1}b_i 2^i \quad a_i,b_i\in\{0,1\}, \quad a\cdot b=\sum_{i=0}^{n-1}a_i b_i \left(\mathrm{mod}\,2\right)$. Obviously, the function $W\left(a,b,2^n\right)$ satisfies the conditions above of $W\left(a,b,q\right)$. In fact, by using $W\left(a,b,2^n\right)$, in the Simon algorithm $W\left(s,c,2^n\right)=1, s\cdot c=0$. Next, let us consider the case when $W\left(a,b,q\right)$ is a discrete Walsh function. We will show that the function $W\left(a,b,2^n\right)$ is a kind of discrete Walsh function.

First, we describe Walsh functions and discrete Walsh functions. We define a function

$$r\left(x\right):\mathbb{R}\to\{-1,1\} \text{ by } r\left(x\right)=\begin{cases}1, & 0\le x<\frac{1}{2}\\ -1, & \frac{1}{2}\le x<1\end{cases}$$

and $r\left(x+1\right)=r\left(x\right)$. Moreover, let $r_l\left(x\right)$ be a function $r_l\left(x\right)=r\left(2^l x\right)$ where $x\in\mathbb{R}, l\in\mathbb{N}$. Then, a *Walsh function* (more precisely, *Walsh-Paley function*) $W_l\left(x\right)$ is defined by $W_l\left(x\right)=\prod_{k=0}^{\infty}r_k\left(x\right)^{l_k}$, where $l=\sum_{k=0}^{\infty}l_k 2^k, l_k\in\{0,1\}$.

Remark. Every value of the function $W_l\left(x\right)$ is always a finite value because $l_k=0$ for k that is sufficiently large (in fact, each value of the Walsh-Paley function is either 1 or -1). Moreover, when let $a\in\mathbb{N}$, $b\in\mathbb{N}$, and $q\in\mathbb{Z}^+$, a *discrete Walsh-Paley function* is defined by $W_a\left(\frac{b}{q}\right)$.

Example. Now, let $q=2^n\ \left(n\in\mathbb{N}\right)$. Then, the function $W_a\left(\frac{b}{q}\right)$ satisfies the following properties [so that this function satisfies the properties of $W\left(a,b,q\right)$]:

$W_0\left(\frac{b}{q}\right)=1$	$W_{a\oplus c}\left(\frac{b}{q}\right)=W_a\left(\frac{b}{q}\right)W_c\left(\frac{b}{q}\right)$
$W_a\left(\frac{b}{q}\right)=W_b\left(\frac{a}{q}\right)$	$\sum_{c=o}^{q-1}W_a\left(\frac{c}{q}\right)W_c\left(\frac{b}{q}\right)=q\delta_{ab}$

.

Namely, a matrix U_{P_q} whose the a-th row and b-th column element o is $\frac{1}{\sqrt{q}}W_a\left(\frac{b}{q}\right)$ is a unitary matrix. The transform $P_q:\left|a\right\rangle\xrightarrow{P_q}\frac{1}{\sqrt{q}}\sum_{c=0}^{q-1}W_a\left(\frac{c}{q}\right)\left|c\right\rangle$ correspond to the matrix U_{P_q} and can be computed in polynomial time of n.

Example. Let H_q be a q-dimensional Hadamard matrix$\left(q=2^n\right)$, that is, when $H_2=\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ then $H_q=H_2\otimes H_{q/2}=\begin{pmatrix}H_{q/2} & H_{q/2}\\ H_{q/2} & -H_{q/2}\end{pmatrix}$. The QTM can execute the matrix $\frac{1}{\sqrt{q}}H_q$ in $O\left(n\right)$ time. Further,

when let H_{kj} be the k-th row and j-th column element of H_q then $H_{kj} = W_{br(k)}\left(\frac{j}{q}\right)$ where for $k = \sum_{i-0}^{n-1} k_i 2^i \left(k_i \in \{0,1\}\right), br(k) = \sum_{i-0}^{n-1} k_{n-i-1} 2^i$ (for a bit string, its reverse order is obtained). The QTM can also execute this procedure in $O(n)$ time. Consequently, the QTM can execute the transform P_q in $O(n)$ time.

Relationships among some Walsh transforms. For the value $a = \sum_{i=0}^{n-1} a_i 2^i$, $a_i \in \{0,1\}$, let

$$g_i \left(i = 0,1,\ldots,n-1\right) = \begin{cases} g_{n-1} = a_{n-1} \\ g_i = a_{i+1} \oplus a_i \left(0 \le i \le n-2\right) \end{cases}.$$

Gray code $g(a)$ of a is defined by $g(a) = \sum_{i=0}^{n-1} g_i 2^i$. Now, when $W\left(a, \frac{b}{q}\right)$ and $H_a\left(\frac{b}{q}\right)$ are discrete *Walsh-Paley* function and *Walsh-Hadamard* function, respectively, the relationships among Walsh functions are $W\left(a, \frac{b}{q}\right) = W_{g(a)}\left(\frac{b}{q}\right)$ and $H_a\left(\frac{b}{q}\right) = W_{br(a)}\left(\frac{b}{q}\right)$.

Moreover, we can describe them by

$W\left(a, \frac{b}{q}\right) = (-1)^{br(g(a))\cdot b}$	$W_a\left(\frac{b}{q}\right) = (-1)^{br(a)\cdot b}$	$H_a\left(\frac{b}{q}\right) = (-1)^{a\cdot b}$