УДК 512.6, 517.9, 519.6

STRUCTURE DESIGN TOOLKIT OF QUANTUM ALGORITHMS. PT 3.

Reshetnikov Andrey[1], Tyatyushkina Olga[2], Ulyanov Sergey[3], Degli Antonio Giovanni[4]

[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;*
Polo Didattico e di Ricerca di Crema;
Via Bramante, 65-26013, Crema (CR), Italy;
e-mail: gda@dsi.unimi.it.

The universality of the quantum Fourier transform in forming the basis of quantum computing algorithms is considered. The unique universal fundamental properties of quantum computing concerning quantum superposition, entanglement and interference are all explicitly represented in terms of quantum multi-particle interferometry.

Keywords: Quantum computing, universal quantum gates, quantum operators, matrix transformation

ИНСТРУМЕНТАРИЙ ПРОЕКТИРОВАНИЯ КВАНТОВЫХ АЛГОРИТМОВ. Ч. 3.

Решетников Андрей Геннадьевич[1] , Тятюшкина Ольга Юрьевна[2], Ульянов Сергей Викторович[3], Джиованни дели Антонио[4]

[1] *Кандидат технических наук, 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: tyatyushkina@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, профессор;*
Поло дидаттико, Крема, факультет информационных технологий;
Виа Браманте, 65-26013, Крема, Италия;
e-mail: gda@dsi.unimi.it.

Рассмотрена универсальность квантового преобразования Фурье в формировании основ разработки структур квантовых вычислительных алгоритмов. Универсальные фундаментальные свойства квантовых вычислений, касающиеся квантовой суперпозиции, запутывания и интерференции, представлены в терминах квантовой многочастичной интерферометрии.

Ключевые слова: квантовые вычисления, универсальные квантовые ячейки, квантовые операторы, матрицы преобразования.

Introduction

Let us consider the description of Quantum Fourier Transform (QFT). The universality of the QFT in forming the basis of quantum computing algorithms is considered. The unique universal fundamental properties of quantum computing concerning quantum superposition, entanglement and interference are all explicitly represented in terms of quantum multiparticle interferometry [1-18].

The Universality of the Quantum Fourier Transform in Forming the Basis of Quantum Computing Algorithms

The *Quantum Fourier Transform* (QFT) on the additive group of integers modulo 2^m is defined by.

$F_{2^m}\left(\lvert a\rangle\right) \;=\; \sum_{y=0}^{2^m-1} e^{(2\pi iay)/2^m}\lvert y\rangle, \quad (1)$	For $a \in \{0,1,2,...,2^m-1\}$.

QFT plays a significant role in the development of the quantum computer (QC). One may note, for example, that the potentially powerful integer factoring algorithm by P. Shor relies critically on the QFT for the detection of periodicity springing from the prime factors.

We can further analyze (1) as follows. First, write

$$\boxed{a = a_1 2^{m-1} + a_2 2^{m-2} + \cdots + a_{m-1}2^1 + a_m 2^0 = (a_1 a_2 \ldots a_m)}$$

and $y = y_1 2^{m-1} + y_2 2^{m-2} + \cdots + y_{m-1}2^1 + y_m 2^0 = (y_1 y_2 \ldots y_m)$.

Then it is well known that

RHS of (1)	=	$\sum_{y=0}^{2^{m-1}} e^{\left(2\pi iay/2^m\right)}\lvert y_1\cdots y_m\rangle$
	=	$\sum_{y=0}^{2^{m-1}} e^{2\pi i(0.a_m)y_1}\lvert y_1\rangle e^{2\pi i(0.a_{m-1}a_m)y_2}\lvert y_2\rangle\cdots e^{2\pi i(0.a_1a_2\ldots a_m)y_m}\lvert y_m\rangle$
	=	$\left(\lvert 0\rangle + e^{2\pi i(0.a_m)}\lvert 1\rangle\right)\left(\lvert 0\rangle + e^{2\pi i(0.a_{m-1}a_m)}\lvert 1\rangle\right)\cdots\left(\lvert 0\rangle + e^{2\pi i(0.a_1a_2\ldots a_m)}\lvert 1\rangle\right)$.

In the above factorization (or “untangling”), each factor is of the form $\lvert 0\rangle + e^{i\omega}\lvert 1\rangle$.

Remark: Such a state can be produced in two steps:

First, apply the transformation $H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$, where H is known as the Walsh-Hadamard transform, to the state $\lvert 0\rangle$: $H\lvert 0\rangle = \frac{1}{\sqrt{2}}\left(\lvert 0\rangle + \lvert 1\rangle\right)$.

Next, apply the phase shift operator $P(\omega) = \begin{bmatrix}1 & 0\\ 0 & e^{i\omega}\end{bmatrix}$, yielding $P(\omega)\left[H\lvert 0\rangle\right] = \frac{1}{\sqrt{2}}\left(\lvert 0\rangle + e^{i\omega}\lvert 1\rangle\right)$

Remark: The RHSs are equal (apart from a normalization coefficient). Therefore, we see that the constituents of the QFT are H and $P(\omega)$. From the quantum optics point of view, H is realized by a half-silvered mirror (beam splitter) and $P(\omega)$ represents a phase shifter, as in a standard Mach-Zehnder interferometer.

First, we wish to emphasize that the QFT strictly by itself is *not universal* in quantum computing (see Remark below). Thus, the question becomes whether the two constituents H and $P(\cdot)$ of QFT are universal or not. We will discuss the following problem:

"*Any QC algorithm can be represented as a composition of Walsh-Hadamard transforms and associated conditional phase shifts*."

Remark: The implication of this problem is that the realization of any QC algorithm translates into a combination of elementary quantum interferometric operations, i.e., single particle beam splitter (Walsh-Hadamard transform) followed by a conditional phase shift. Any QC algorithm can thus be formulated, or reformulated, in terms of elementary multiparticle quantum interferometric operations. The unique universal fundamental properties of QC concerning quantum superposition, entanglement and interference are all explicitly represented in terms of quantum multiparticle interferometry (QMI).

Remark. QMI practically is not to be taken as a proposed embodiment of a QC any more than the Turing machine is to be taken as a literal construction in classical computing. Rather, Ekert has suggested its equivalence to QC in the sense of its universality, meaning that QMI could be viewed as the closest QC analogue of the classical Turning machine (through the universality theorem established in this appendix). This concept and viewpoint should provide physical insights into the operational aspects and can facilitate efficient design of a universal QC.

Mathematical proof of the Universality of H and P(.)

As usual, we let $U(n)$ to denote the unitary group on n-dimensional space. By abuse of notation, we regard $U(n)$ the same as the multiplicative group of all $n\times n$ orthogonal matrices.

$SO(n)$ denotes the orthogonal group on *n*-dimensional spaces or, equally, the multiplicative group of all $n\times n$ orthogonal matrices. We also define the *maximal tours* $T(n)$ in$U(n)$ as

$$T(n)=\left\{diag\left(e^{i\omega_1},...,e^{i\omega_n}\right)\middle|\omega_1,\omega_2,...,\omega_2\in\mathbb{R}\right\},$$

i.e., $T(n)$ consists of all $n\times n$ diagonal matrices whose diagonal entries are complex numbers of unit magnitude. $T(n)$ is a subgroup of the multiplicative group $U(n)$.

Let A be a collection of $n\times n$ unitary matrices. We will use $g_n(A)$ to denote *the unitary subgroup of* $U(n)$ *generated* by *A*, i.e.,

$$g_n(A)=\bigcap_{\alpha}\left\{G_\alpha\,\middle|G_\alpha \textit{is a subgroup of } U(n), A\subseteq G_\alpha\right\}$$

We will write $g_n(A)$ simply as $g(A)$ if the value of n is clear from the context.

We begin with $n=2$.

Lemma: We have $U(2)\cong g\left(SO(2),T(2)\right)$, i.e., $U(2)$ is generated by $SO(2)$ and $T(2)$; more precisely, for every $A\in U(2)$, we have

$$A=\begin{bmatrix} e^{i\delta} & 0\\ 0 & e^{i\delta}\end{bmatrix}\begin{bmatrix} e^{i\alpha/2} & 0\\ 0 & e^{-i\alpha/2}\end{bmatrix}\begin{bmatrix} \cos\omega & \sin\omega\\ -\sin\omega & \cos\omega\end{bmatrix}\begin{bmatrix} e^{i\beta/2} & 0\\ 0 & e^{-i\beta/2}\end{bmatrix},$$

for *some* $\alpha,\beta,\delta,\omega\in\mathbb{R}$.

Lemma: $T(2)\subseteq g\left(H,P(\cdot)\right)$.

Proof. We first note that the NOT-gate $X\equiv\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$ can be obtained as

$$X=HP(-\pi)H.$$

Therefore $X\in g\left(H,P(\cdot)\right)$. From this, we have

$$XP(\omega_1)XP(\omega_2)=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & e^{i\omega_1}\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & e^{i\omega_2}\end{bmatrix}=\begin{bmatrix}e^{i\omega_1} & 0\\ 0 & e^{i\omega_2}\end{bmatrix},$$

for any given $\omega_1,\omega_2\in\mathbb{R}$. Therefore $g\left(H,P(\cdot)\right)$ contains the maximal torus $T(2)$.

Lemma $SO(2)\subseteq g\left(H,P(\cdot)\right)$.

Proof. For each rotation matrix

$$R(\omega)=\begin{bmatrix}\cos\omega & \sin\omega\\ -\sin\omega & \cos\omega\end{bmatrix},$$

we easily verify that

$$R(\omega)=P\left(\frac{\pi}{2}\right)HP(\omega)XP(-\omega)HP\left(-\frac{\pi}{2}\right).$$

Theorem: $g\left(H,P(\cdot)\right)\cong U(2)$.

Proof. This follows immediately from Lemmas.

Decomposition Procedure of General Finite Dimensional Unitary Transformations into a Product of Plane Unitary Transformations

First, we define a special type of unitary transformations $T_{pq}(\phi\sigma)\in U(n)$ by $T_{pq}(\phi\sigma)=\lfloor t_{ij}\rfloor_{n\times n}$, $1\le p,q\le n,\quad p\ne q,$ where

$$t_{ij}=\begin{cases}1 & i=j,i\ne p,i\ne q,\\ \cos\phi, & i=j=p\;or\;i=j=q,\\ 0, & i\ne j,i\ne p,j\ne q\;and\;i\ne q,j\ne p,\\ -e^{-i\sigma}\sin\phi, & i=p\;and\;\;j=q,\\ e^{i\sigma}\sin\phi, & i=q\;and\;\;j=p;\end{cases}$$

$$\begin{matrix}\boxed{p} & \boxed{q}\\ \downarrow & \downarrow\end{matrix}$$

$$T_{pq}(\phi,\sigma)=\begin{array}{c}\\ \\ \\ \\ \boxed{p}\}\rightarrow \\ \boxed{q}\}\rightarrow \\ \\ \\ \\ \end{array}\begin{bmatrix}1 & 0 & 0 & & & & & 0\\ 0 & 1 & & & & & & \\ & & 1 & & & & & \\ & & & \ddots & & & & \\ & & & & \cos\phi & -e^{-i\sigma}\sin\phi & & \\ & & & & e^{i\sigma}\sin\phi & \cos\phi & & \\ & & & & & & 1 & 0\\ 0 & 0 & & & & & \ddots & \\ & & & & & & 0 & 1\end{bmatrix}$$

$T_{pq}(\phi,\sigma)$ is just a plane unitary transformation acting non-travially only on states *p* and *q*.

Let $V\in U(n)$. We want to find some $T_{n,n-1}(\phi,\sigma)$ such that $T^{*}_{n,n-1}V=V'=\left[\upsilon'_{ij}\right]_{n\times n}$, where $\upsilon'_{n-1,n}=0$:

$$T^{*}_{n,n-1}V=\begin{bmatrix}1 & 0 & 0 & & & \\ 0 & 1 & 0 & & & \\ 0 & 0 & 1 & & \mathrm{O} & \\ & & & \ddots & & \\ & & & & \cos\phi & e^{-i\sigma}\sin\phi\\ & & \mathrm{O} & & -e^{i\sigma}\sin\phi & \cos\phi\end{bmatrix}\begin{bmatrix}\upsilon_{11} & \cdots & \upsilon_{1,n-1} & \upsilon_{1n}\\ \vdots & & \vdots & \vdots\\ \upsilon_{n-1,1} & \cdots & \upsilon_{n-1,n-1} & \upsilon_{n-1,n}\\ \upsilon_{n1} & \cdots & \upsilon_{n,n-1} & \upsilon_{nn}\end{bmatrix},$$

so $\upsilon'_{n-1,n}=\upsilon_{n-1,n}\cos\phi+\upsilon_{nn}e^{-i\sigma}\sin\phi.$

We consider all possibilities:

Case 1:	$\upsilon_{n-1,n}=0.$ Then we choose $\phi=0,\sigma=0,$ i.e., $T_{n-1,n}(\phi,\sigma)=I_n$, and we obtain $\upsilon'_{n-1,n}=\upsilon_{n-1,n}=0.$
Case 2:	$\upsilon_{n-1,n}\neq 0,\upsilon_{nn}=0.$ Then choose $\phi=\pi/2,\sigma=0.$ Obtain $\upsilon'_{n-1,n}=0.$
Case 3:	$\upsilon_{n-1,n}\neq 0,\upsilon_{nn}\neq 0.$ Write $\upsilon_{n-1,n}=r_{n-1,n}e^{i\theta_{n-1,n}},\upsilon_{nn}=r_{nn}e^{i\theta_{nn}}.$ Choose $\sigma=-\theta_{n-1,n}+\theta_{nn}$ and $\phi=\tan^{-1}\left(-r_{n-1,n}/r_{nn}\right).$ Obtain $\upsilon'_{n-1,n}=\cos\phi\cdot r_{n-1,n}e^{i\theta_{n-1,n}}+\sin\phi\cdot r_{nn}e^{i(-\sigma+\theta_{nn})}$ $=\left(\dfrac{r_{n-1,n}}{r_{nn}}+\tan\phi\right)r_{nn}\cos\phi e^{i\theta_{n-1,n}}=0.$

Therefore, we have found $T_{n,n-1}\in U(n)$ such that

$$T^{*}_{n,n-1}V=\begin{bmatrix}* & \cdots & * & \upsilon'_{1n}\\ \vdots & & \vdots & \vdots\\ * & \cdots & * & \upsilon'_{n-2,n}\\ & & & 0\\ \upsilon'_{n1} & \cdots & \upsilon'_{n,n-1} & \upsilon'_{nn}\end{bmatrix}.$$

Similarly, we can find $T_{n,n-2},T_{n,n-3},\ldots,T_{n,1}$ such that

$$T^*_{n,n-2}T^*_{n,n-1}V=\begin{bmatrix} * & & * & \upsilon''_{1n} \\ \vdots & & \vdots & \vdots \\ \vdots & & \vdots & \upsilon''_{n-3,n} \\ & & & 0 \\ * & & * & 0 \\ \upsilon''_{n1} & \dots & \upsilon''_{n,n-1} & \upsilon''_{nn} \end{bmatrix},$$

$$\vdots$$

$$T^*_{n1}T^*_{n2}\dots T^*_{n,n-2}T^*_{n,n-1}V=\begin{bmatrix} * & & * & 0 \\ \vdots & & \vdots & 0 \\ \vdots & & \vdots & \vdots \\ * & & * & 0 \\ \tilde{\upsilon}_{n1} & \dots & \tilde{\upsilon}_{n,n-1} & \tilde{\upsilon}_{nn} \end{bmatrix}\equiv W.$$

Since W is unitary, we conclude $\tilde{\upsilon}_{n1}=\tilde{\upsilon}_{n2}=\cdots=\tilde{\upsilon}_{n,n-1}=0$ and $\tilde{\upsilon}_{nn}=e^{i\alpha_n}\equiv d_n$ for some $\alpha_n\in\mathbb{R}.$ Thus

$$T^*_{n1}T^*_{n2}\dots T^*_{n,n-2}T^*_{n,n-1}V=\begin{bmatrix} & & & 0 \\ & \underline{**}| & & \vdots \\ & & & 0 \\ 0 & \dots & 0 & d_n \end{bmatrix}.$$

Now, applying the same technique to the remaining $(n-1)\times(n-1)$ undiagonalized matrix block $(**)$ above, together with a simple induction argument, we obtain plane unitary transformation $T_{n1},\dots,T_{n,n-1},T_{n-1,1},\dots,T_{n-1,n-2},\dots,T_{31},T_{32}$ and T_{21} such that

$$T^*_{21}T^*_{31}T^*_{32}T^*_{41}\dots T^*_{n-1,1}\dots T^*_{n-1,n-2}T^*_{n1}\dots T^*_{n,n-1}V=\begin{bmatrix} d_1 & & \\ & d_2 & & 0 \\ & & \ddots & \\ & 0 & & d_n \end{bmatrix}=D,$$

where $d_j=e^{i\alpha_j}$ for $j=1,2,\dots,n.$

Therefore

$$\boxed{\begin{aligned} V&=T_{n,n-1},T_{n,n-2}\dots T_{n1}T_{n-1,n-2}\dots T_{n-1,1}\dots T_{32}T_{31}T_{21}D \\ &=\left(\prod_{i=1}^{n}\prod_{j=1}^{i-1}T_{i,j}\right)D. \end{aligned}}$$

At this point, it should already be clear that $U(2^n)$ *can be generated through controlled-*$U(2)$ *gates*, for any n = 1,2,…. Let us give the following concise, rigorous treatment as to how to construct any $V\in U(2^n)$ from a serial connection of a collection of unitary matrices V_{ij}, where each V_{ij} is a (generalized) controlled-$U(2)$ gate. The precise statement is given below.

Theorem: Let $V\in U(2^n)$. Then

$$V=\prod_{i=1}^{2^n-1}\prod_{j=0}^{i-1}V_{ij}. \qquad (2)$$

For a collection of matrices $V_{ij} \in U\left(2^n\right)$ such that

$$\left\{\begin{matrix} V_{ij}: \quad S_{ij} \to S_{ij} \quad \textit{is the identity transformation}, \\ S_{ij} \equiv span\left\{|m\rangle \middle| m \in \{0,1,\ldots 2^n-1\}, m \neq i, m \neq j\right\} \\ 0 \le j < i \le 2^n - 1. \end{matrix}\right\} \tag{3}$$

In other words, each $V \in U\left(2^n\right)$ is a product of (generalized) controlled-U(2) unitary matrices V_{ij}, which acts nontrivially only on $S_{ij}^{\perp} = span\{|i\rangle, |j\rangle\}$.

Proof. We first quote the following fact (see above). For any $V \in U\left(2^n\right)$, there exists a collection of unitary matrices $T_{i,j}, 0 \le j < i \le 2^n - 1$, and a $D \in T\left(2^n\right)$ such that $V = \left(\prod_{i=1}^{2^n-1}\prod_{j=0}^{i-1} T_{i,j}\right) D$, where $T_{i,j} \in SO\left(2^n\right) \subseteq U\left(2^n\right)$ is a rotation involving $|i\rangle$ and $|j\rangle$ and satisfying the above mentioned condition. Now we can break up D into

$$D = \begin{pmatrix} d_0 & & & \\ & d_1 & & \\ & & \ddots & \\ & & & d_{2^n-1} \end{pmatrix} = D_1 D_2 \ldots D_{2^n-1}$$

$$\text{where } D_1 = \begin{pmatrix} d_0 & 0 & & & \\ 0 & d_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \text{ and } D_i = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & d_i & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}$$

for $i = 2,3,\ldots,2^n - 1$. It is easy to see that D_1 acts trivially except on $|0\rangle$ and $|1\rangle$, and the other D_i's act non-trivially only on $|i\rangle$. In addition, D_i's commute with each other, and each D_i commutes with $T_{k,l}, \forall 0 \le l < k < i$ as well.

Thus,

$$\begin{aligned} V &= T_{2^n-1,2^n-2} \ldots T_{2^n-1,0} T_{2^n-2,2^n-3} \ldots T_{2^n-2,0} \ldots T_{2,1} T_{2,0} T_{1,0} D_1 D_2 \ldots D_{2^n-1} \\ &= \left.\begin{matrix} T_{2^n-1,2^n-2} T_{2^n-2,2^n-3} \ldots T_{2^n-1,0} D_{2^n-1} \\ T_{2^n-2,2^n-3} \ldots T_{2^n-2,0} D_{2^n-2} \\ \ldots\ldots \\ T_{2,1} T_{2,0} D_2 \\ T_{1,0} D_1 \end{matrix}\right\} 2^n - 1 \end{aligned}$$

strings of products. For $0 \le j < i \le 2^n - 1$, define

$$V_{ij} = \begin{cases} T_{i,j} & \textit{if } \; j \neq 0, \\ T_{i,j} D_i = T_{i,0} D_i & \textit{if } \; j = 0. \end{cases}$$

Therefore we have reached $V = \prod_{i=1}^{2^n-1}\prod_{j=0}^{i-1} V_{ij}$, where each V_{ij} is a unitary matrix which acts nontrivially only on the states $|i\rangle$ and $|j\rangle$ satisfying the above mentioned expression.

Remark. The factoring of *D* into the product of $D_1, D_2, \ldots$ and D_{2^n-1} in the presented forms is peculiar in the sense that D_1 is chosen differently from the other D_i's, $i \neq 1.$ It must be done this way. The reason for this is that there are $2^n - 1$ strings of products as indicated above. Therefore *D* must be factorized to have $2^n - 1$ factors $D_1, D_2, \ldots, D_{2^n-1}$, in the unique way.

Remark. Now it can be readily seen that the QFT itself is not universal in the sense that $U(2^n)$ is not generated by F_{2^n} (cf., with case $m = n$ therein) or (generalized) controlled-F_{2^m} (where $m < n$) operations. First, check $n = 1:$ we see that $F_{2^n} = F_2$ is actually the Walsh-Hadamard transform *H* (apart from the normalization factor $1/\sqrt{2}$). Therefore, the phase shifts $P(\omega)$ in (2) cannot be generated by F_2 because $P(\omega)$ has eignevalues 1 and $e^{i\omega}$ while *H* has eignevalues 1 and –1. For a general positive integer *n*, the range of F_{2^n} or of controlled-$F_{2^m}, m < n,$ consists at most of linear combinations of states of the form $e^{2\pi i[(0.a)y_1 + (0.a_{n-1}a_n)y_2 + \cdots + (0.a_1 \ldots a_n)y_n]}|y_1 \ldots y_n\rangle,$ where $a_j, y_j \in \{0,1\},$ for $j = 1,2,\ldots,n.$

The phases of such states are *not even dense* with respect to all possible phases $e^{2\pi i\theta}$, $0 \leq \theta < 2\pi.$

Remarks on Circuits. The decomposition (3) is a mathematical rendering of above mentioned statement and answers the conjecture affirmatively.

Each factor V_{ij} in (3) satisfies (4) and thus V_{ij} acts nontrivially only on the states $|i\rangle$ and $|j\rangle.$ Denote the restriction of V_{ij} to the 2-dimensional subspace $\varsigma_{ij}^{\perp} = span\{|i\rangle, |j\rangle\}$ by $V_{ij}.$ Then $\hat{V}_{ij} \in U(2).$ Each V_{ij} is not a standard $\Lambda_{n-1}(\hat{V}_{ij})$ gate is the sense that the *controls are states rather than bits.*

Nevertheless, point out how to rearrange basis states with a "gray code connecting state $|i\rangle$ to state $|j\rangle$" such that V_{ij} becomes unitarily equivalent to $\Lambda_{n-1}(\hat{V}_{ij}).$ In this sense, V_{ij} are generalized controlled-$\hat{V}_{ij}$ gates.

Proposition. *The symmetric group* S_{2^n} of *permutations on the symbols* 0,1,2,…,2 ^{n} -1 *is generated by the 2-cycle* $(2^n - 2, 2^n - 1)$ *and the* 2 ^{n} *-cycle* $(0,1,2,\ldots,2^n - 1).$

Proof. This is a basic fact of group theory.

Incidentally, we note that the 2-cycle $(2^n - 2, 2^n - 1)$ is a permutation between the states $\left|\underbrace{11\ldots10}_{n\,bits}\right\rangle$ and $\left|\underbrace{11\ldots1}_{n\,bits}\right\rangle$ and thus can be realized by the controlled-NOT gate with the *n*th qubit as the *target bit* and the first $(n-1)$ bits as the *control bits* as shown in Fig. 1.

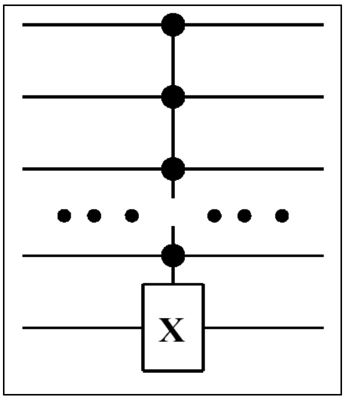

Figure 1: The n -bit controlled-NOT gate

On the other hand, the 2^n-cycle $(0,1,2,...,2^n-1)$ makes the rotation of the states $|0\rangle \to |1\rangle \to \cdots |2^n-2\rangle \to |2^n-1\rangle \to |0\rangle$, i.e., the $|x\rangle \to |x+1 \bmod 2^n\rangle$ operation. This can be implemented by the circuit as shown in Fig. 2.

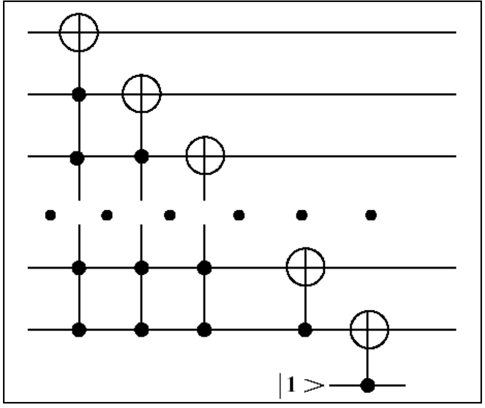

Figure 2. This circuit implements the operation $|x\rangle \to |x+1 \bmod 2^n\rangle$ or, equivalently, the 2^n -cycle (0, 1, 2, . . . , 2^n . 1) in Proposition. Note that the bit $|1\rangle$ at the bottom of the figure is the “scratch bit” which is sometimes omitted in circuit drawing. All the gates in this circuit are controlled-NOT gates

Therefore, any permutation of the basis states $|x\rangle, x = 0,1,2,\ldots,2^n-1$, can be realized by finitely many controlled-NOT operations consisting of circuits as shown in Figs 1 and 2.

Thus, each factor V_{ij} in (2) can be realized by the circuit as shown in Fig. 3.

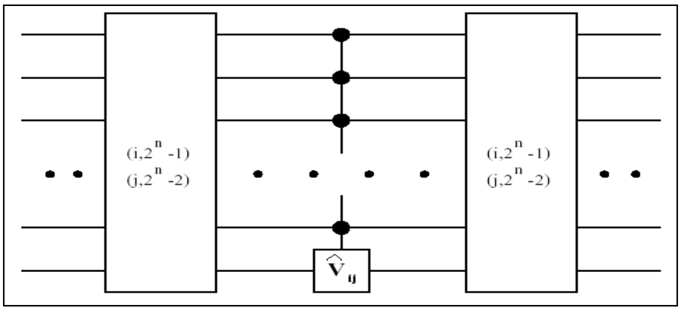

Figure 3: The unitary matrix V_{ij} as a controlled-$\hat{V}_{ij}$ gate where $\hat{V}_{ij} \in U(2)$. The operations $(i, 2^n - 2)$ and j, 2n . 2) in the two boxes are cyclic permutations (which can be realized by concatenations of circuits in Figs 2 and 3

By concatenating together all the blocks V_{ij} as shown in Fig. 2 according to the factorization (2), we have constructed all $V \in U(2^n)$ with controlled-$\hat{V}_{ij}$ gates according to (2). Each $\hat{V}_{ij} \in U(2^n)$ is then further formed from concatenations of the gates $H, P(\omega) \in U(2)$ according to corresponding Theorem. It is in this sense that we have the universality of the Walsh-Hadamard gate *H* and the phase shift gate $P(\cdot)$ and, consequently, that of the QFT with the affirmative answer to the above question.

Example: *Another Way to Perform the Quantum Fourier Transform in Linear Parallel Time*. Shor's factoring algorithm suggests that quantum computers can do things in polynomial time that classical computers cannot. However, since decoherence due to storage errors is a function of time, we should also ask to what extent we can parallelize quantum algorithms; if we can do many quantum operations at once, rather than serially, we can solve larger problems before our computer decoherens.

Consider a quantum circuit operating on a set of qubits, containing one-qubit gates (2×2 unitary matrices) and the two-qubit controlled-not-gate; these are universal for quantum computation. We can define the *depth* of this circuit as the number of layers, where each layer consists of gates operating on mutually disjoint sets of qubits; that is, each qubit interacts with at most one other qubit at time. (In a model of quantum computation where one qubit can simultaneously interact with several others, we could allow gates operating on the same qubit in the same level, as long as these gates all mutually commute.)

The heart of Shor's algorithm is the *Quantum Fourier Transform.* If we represent *n*-digit numbers $|a\rangle$ with *n* qubits, the QFT maps $|a\rangle$ to $2^{-n/2}\sum_{b=0}^{2^n-1} e^{2\pi i ab/2^n}|b\rangle$.

We exhibit a circuit with depth $O(n)$ for performing the QFT.

Griffiths and *Niu* have already done this, in fact in a more natural way.

We exhibit a quantum circuit that performs the QFT on *n* qubits in $O(n)$ depth. Thus, a parallel quantum computer can carry out the QFT in linear time. Griffiths and Niu have already shown this. We also speculate as to whether the QFT might be in the class QNC of problems solvable in logarithmic parallel time.

The standard quantum algorithm for the QFT takes $n(n-1)/2$ gates. One way to construct it is to reshuffle the rows of the matrix by putting the digits of the input in reverse order. Then for *n*=3, for instance, we have

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i & 1 & i & -1 & -i \\ 1 & -i & -1 & i & 1 & -i & -1 & i \\ 1 & e^{\pi i/4} & i & e^{3\pi i/4} & -1 & e^{5\pi i/4} & -i & e^{7\pi i/4} \\ 1 & e^{5\pi i/4} & i & e^{7\pi i/4} & -1 & e^{\pi i/4} & -i & e^{3\pi i/4} \\ 1 & e^{3\pi i/4} & -i & e^{\pi i/4} & -1 & e^{7\pi i/4} & i & e^{5\pi i/4} \\ 1 & e^{7\pi i/4} & -i & e^{5\pi i/4} & -1 & e^{3\pi i/4} & i & e^{\pi i/4} \end{pmatrix}$$

where we are suppressing a factor of $2^{-3/2}$.

If we call this *F*(3), we immediately notice that its upper-left and upper-right quadrants are

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \\ 1 & -i & -1 & i \end{pmatrix}$$

which is simply $F(2)$. The lower-left and lower-right quadrants of $F(3)$ are $F(2)$ and $-F(2)$, with a series of phase shifts applied to the columns; this can be expressed by multiplying on the right by the matrix

$$\begin{pmatrix} 1 & & & \\ & e^{\pi i/4} & & \\ & & i & \\ & & & e^{3\pi i/4} \end{pmatrix}$$

which we will call M. In general, we can write

$F(n+1)$	=	$\frac{1}{\sqrt{2}}\begin{pmatrix} F & F \\ FM & -FM \end{pmatrix}$
	=	$\begin{pmatrix} F & \\ & F \end{pmatrix}\cdot\begin{pmatrix} 1 & \\ & M \end{pmatrix}\cdot\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

We recognize this as the circuit for $F(n)$ applied to the n least significant qubits, followed by a gate where the most significant qubit controls whether or not to apply the phase shifts M , followed by the *Hadamard operator* $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ applied to the most significant qubit.

Finally, note that M is simply a tensor product of independent one-qubit operations

$$M = \begin{pmatrix} 1 & \\ & i \end{pmatrix} \otimes \begin{pmatrix} 1 & \\ & e^{\pi i/4} \end{pmatrix} \otimes \begin{pmatrix} 1 & \\ & e^{\pi i/8} \end{pmatrix} \otimes \cdots$$

Then the controlled-M gate becomes a series of controlled phase-shift gates

$$\begin{pmatrix} 1 & \\ & M \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & i \end{pmatrix} \otimes \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{\pi i/4} \end{pmatrix} \otimes \cdots$$

These gates are symmetric, in that the “controlled” and “controlling” qubits are interchangeable. Putting all this together gives us the recursive construction.

To what extent can this circuit be parallelized?

Even though all the phase shift gates within a given pair of H’s commute with each other, we can’t perform them simultaneously unless we can couple one qubit to multiple qubits at the same time, and they don’t commute with the H preceding them. Thus, it would appear that all $O(n^2)$ gates have to be applied in series.

However, we can turn this circuit onto one where most of the gates commute, so that many can be performed simultaneously, in the following way. Note that H is its own inverse. Conjugating a phase shift gate with H gives

$H\begin{pmatrix} 1 & \\ & e^{i\theta} \end{pmatrix}H$	=	$\frac{1}{2}\begin{pmatrix} 1+e^{i\theta} & 1-e^{i\theta} \\ 1-e^{i\theta} & 1+e^{i\theta} \end{pmatrix}$

Call this matrix R_θ. Then if we pass the H operators through the phase shifts to the right, we get the circuit, where the controlled phase-shift gates have been replaced by controlled-R_θ gates.

Now note that two controlled-R_θ gates commute in every case except when the ‘control’ of one is the ‘controlled’ qubit of the other. Formally, if R_{ij} is a controlled-R_θ gate with qubit I controlling qubit j, then R_{ij} and R_{kl} commute unless $j=k$ or $i=l$. We can perform commuting gates simultaneously, as long as we respect the ordering between pairs of this kind. Adding the constraint that each qubit only interact with one other in each layer gives the circuit with the depth $2n-2$, linear in *n*.

It is easy to show that $2n-2$ is the minimal depth for this set of gates. We have one gate R_{ij} for every pair i<j, and R_{ij} must be performed after R_{jk}. Therefore, two gates R_{ij} and R_{kl} cannot be in the same layer if i < j < k < l, since R_{jk} has to precede R_{ij} but follow R_{kl}. This means that the $n-1$ gates R_{ij} where $j=i+1$ must all be in separate levels; since each qubit can only interact with one gate per layer, the $n-2$ gates R_{ij} where $j=i+2$ also need their own layers. Adding this to a final layer of *H*'s gives depth $2n-2$.

Remark. Of course, this does not mean that a different set of gates couldn't solve the QFT more efficiently. It would be especially nice if the QFT could be accomplished by a quantum circuit with depth *O*(log *n*). This would put it in QNC^1, the quantum analog of the class NC^1 of problems solvable in logarithmic time by a parallel computer. We would also add the requirement that only a polynomial number of 'ancilla' qubits be used, corresponding to a polynomial number of processors.

How would this be done?

Each qubit controls and receives phase shifts on and from $O(n)$ other qubits. We can easily 'fan out' $O(n)$ copies of each controlling qubit with a reversible circuit of depth *O*(log *n*) consisting of controlled-not gates. Classically, we could 'fan in' *n* phase shifts on a given qubit in depth *O*(log *n*) by composing them in pairs.

However, it does not seem to be so easy to combine quantum gates in this way. We need some representation of phases so that they can be added in pairs with a linear, unitary operator.

In one case, a quantum circuit can be parallelized by re-writing its gates, and lumping them into mutually commuting groups that can be performed simultaneously.

Toffoli and Control-NOT in universal quantum computation.

A set of quantum gates G (also called a basis) is said to be universal for quantum computation if any unitary operator can be approximated with arbitrary precision by a circuit involving only those gates (called a G-circuit). Since complex numbers do not help in quantum computation, we also call a set of real gates universal if it approximates arbitrary real orthogonal operators.

Which set of gates is universal for quantum computation?

This basic question is important both in understanding the power of quantum computing and in the physical implementations of quantum computers, and has been studied extensively.

Examples of universal bases are: (1) Toffoli, Hadamard, and $\frac{\pi}{4}-$gate, due to Kitaev; (2) CNOT, Hadamard, and $\frac{\pi}{8}-$gate, due to Boykin, Mor, Pulver, Roychowdhury, and Vatan; and (3) CNOT plus the set of all single-qubit gate, due to Barenco, Bennett, Cleve, DiVincenzo, Margolus, Shor, Sleator, Smolin, and Weinfurter.

Another basic question in understanding quantum computation is:

Where does the power of quantum computing come from?

Motivated by this question, we rephrase the universality question as follows:

Suppose a set of gates G already contains universal classical gates, and thus can do universal classical computation, what additional quantum gate(s) does it need to do universal quantum computation? Are there some gates that are more "quantum" than some others in brining more computational power?

What additional gates are needed for a set of classical universal gates to do universal quantum computation? We answer this question by proving that any single-qubit real gate suffices, except those that preserve the computational basis.

The result of Gottesman and Knill implies that any quantum circuit involving only the Control-NOT and Hadamard gates can be efficiently simulated by a classical circuit. In contrast, Control-NOT plus any single-qubit real gate that does not preserve the computational basis and is not Hadamard (or its alike) are universal for quantum computing.

Previously only a “generic” gate, namely a rotation by an angle incommensurate with π , is known to be sufficient in both problems, if only one single-qubit gate is added.

Without loss of generality, we assume that G contains the Toffoli gate, since it is universal for classical computation. The above three examples of universal bases provide some answers to this question. It is clear that we need at least one additional gate that does not preserve the computational basis. Let us call such a gate basis changing. The main result is that essentially the basis-changing condition is the only condition we need:

Theorem: The Toffoli gate and any basis-changing single-qubit real gate are universal for quantum computing.

Remark. The beautiful Gottesman-Knill Theorem implies that any circuit involving CNOT and Hadamard only can be simulated efficiently by a classical circuit. It is natural to ask what if Hadamard is replaced by some other gate. We know that if this replacement R is a rotation by an irrational (in degrees) angle, then R itself generates a dense subset of all rotations, and thus is universal together with CNOT, by Barenco et al. What if the replacement is a rotation of rational angles? We show that Hadamard and its alike are the only exceptions for a basis-changing single-qubit real gate, in conjunction with CNOT, to be universal.

Theorem: Let T be a single-qubit real gate and T^2 *does not preserve the computational basis. Then* $\{CNOT, T\}$ *is universal for quantum computing*.

A basis is said to be complete if it generates a dense subgroup of $U(k)$ modular a phase, or $O(k)$ for some $k \geq 2$. Each of the two bases in the above theorems gives rise to a complete basis. By the fundamental theorem of Kitaev and Solovay, any complete basis can efficiently approximate any gate (modular a phase), or real gate if the basis is real. Therefore, any real gate can be approximated with precision ε using polylog$\left(\frac{1}{\varepsilon}\right)$ gates from either basis, and any circuit over any basis can be simulated with little blow-up in the size.

We also provide an alternative prove for Theorem by directly constructing the approximation circuit for an arbitrary real single-qubit gate, instead of using Kitaev-Solovay theorem. The drawback of this construction is that the approximation is polynomial in $\frac{1}{\varepsilon}$; however, it is conceptually simpler, and uses some new idea that does not seem to have appeared before (for example, in the approximation for Control-sign-flip).

There is a broader concept of universality based on computations on encoded qubits, that is, fault-tolerant quantum computing.

Preliminary. Denote the set $\{1, 2, \cdots, n\}$ by $[n]$. The (pure) state of a quantum system is a unit vector in its state space. The state space of one quantum bit, or qubit, is the two dimensional complex Hilbert space, denoted by H. A pre-chosen orthonormal basis of H is called the computational basis and is denoted by $\{|0\rangle, |1\rangle\}$.

The state space of a set of n qubits is the tensor product of the state space of each qubit, and the computational basis is denoted by

$$\{|b\rangle = |b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_n\rangle : b = b_1 b_2 \cdots b_n \in \{0,1\}^n\}$$

A gate is a unitary operator $\bigcup \in U(H^{\otimes r})$, for some integer $r > 0$. For an ordered subset A of a set of n qubits, we write $\bigcup[A]$ to denote applying $\bigcup$ to the state space of those qubits. A set of gates is also called a basis. A quantum circuit over a basis G, or a G circuit, on n qubits and of size m is a sequence

$\bigcup_1[A_1],\bigcup_2[A_2],\cdots,\bigcup_m[A_m],$ where each $\bigcup_i\in G$ and $A_i\subseteq[n]$. Sometimes we use the same notation for a circuit and for the unitary operator that it defines.

Definition: The operator $\bigcup := H^{\otimes r}\to H^{\otimes r}$ is approximated by the operator $\tilde{\bigcup}: H^{\otimes N}\to H^{\otimes N}$ using the ancilla state $|\Psi\rangle\in H^{\otimes N-r}$ if, for arbitrary vector $|\xi\rangle\in H^{\otimes r}$,

$$\left\|\tilde{\bigcup}\left(|\xi\rangle\otimes|\Psi\rangle\right)-\bigcup|\xi\rangle\otimes|\Psi\rangle\right\|\le\varepsilon\left\||\xi\rangle\right\|.$$

Let G be a basis. A G-ancilla state, or an ancilla state when G is understood, of l qubits is a state $A|b\rangle$, for some *G*-curcuit *A* and some $b\in\{0,1\}^l$. A basis *G* is set to be universal for quantum computing if any gate (modular a phase), or any real gate when each gate in *G* is real, can be approximated with arbitrary precisions by G-circuits using G-ancillae. By a phase, we mean the set $\{\exp(i\alpha):\alpha\in\mathbb{R}\}$. The basis is set to be complete if it generates a dense subgroup of $U(k)$ modular a phase, or $O(k)$ when its real for some $k\ge 2$. A complete basis is clearly universal.

We introduce the standard notations for some gates we shall use later. Denote the identity operator on H by I. We often identify a unitary operator by its action on the computational basis. The Pauli operators σ^x and σ^z, and the Hadamard gate H are

$$\sigma^x:=\begin{pmatrix}0&1\\1&0\end{pmatrix},\quad \sigma^z:=\begin{pmatrix}1&0\\0&-1\end{pmatrix},\quad H:=\frac{1}{\sqrt{2}}\begin{pmatrix}1&1\\1&-1\end{pmatrix}.$$

Example: If $\bigcup$ is a gate on r qubits, for some $r\ge 0$ (when $r=0$, $\bigcup$ is a phase factor), $\Lambda^k(\bigcup)$ is the gate on $k+r$ qubits that applies $\bigcup$ to the last r qubits if and only if the first k qubits are in $|1\rangle^{\otimes k}$. The superscript k is omitted if $k=1$. Changing the control condition to be $|0\rangle^{\otimes k}$, we obtain $\overline{\Lambda}^k(\bigcup)$.

The Toffoli gate is $\Lambda^2(\sigma^x)$, and CNOT is $\Lambda(\sigma^x)$. . Evidently the latter can be realized by the former. From now on we only consider real gates. A gate g is said to be basis-changing if it does not preserve the computational basis.

Completeness proofs.We will introduce the proof of the following theorems, from which Theorem 2 and Theorem 1 follow immediately.

Theorem. Let S be any single-qubit real gate that is basis-changing after squaring.

Then $\{CNOT,S\}$ *is complete*.

Theorem. The set $\{\Lambda^2(\sigma^x),H\}$ *is complete*.

We need the following two lemmas, which fortunately have been proved.

Lemma (Wlodarski). If α is nit an integer multiple of $\pi/4$, and $\cos\beta=\cos^2\alpha$, then either α or β is an irrational multiple of π .

Lemma (Kitaev). Let M be a Hilbert space of dimension ≥ 3, $|\xi\rangle\in M$ a unit vector, and $H\subset SO(M)$ be the stabilizer of the subspace $\mathbb{R}(|\xi\rangle)$. If $V\in O(M)$ does not preserve $\mathbb{R}(|\xi\rangle)$, $H\bigcup V^{-1}HV$ generates a dense subgroup of SO(M).

Proof of Theorem. Define $\bigcup:=\left(S\otimes S\cdot\Lambda(\sigma^x)[1,2]\right)^2$. It suffices to prove that $\bigcup$ and $\Lambda(\sigma^x)$ generate a dense subgroup of SO(4). Without loss of generality, we assume that $\bigcup$ is a rotation by an angle θ , the other case can be proved similarly. The by the assumption, θ is not an integer multiple of $\pi/4$.

Direct calculation shows that $\bigcup$ has eigenvalues $\{1,1,\exp(\pm i\alpha)\}$, where $\alpha=2\arccos\cos^2\theta$

The two eigenvectors with eigenvalue 1 are $\left|\xi_1\right\rangle := \frac{1}{2}\left(\left|00\right\rangle - \left|01\right\rangle + \left|10\right\rangle + \left|11\right\rangle\right)$, and

$$\left|\xi_2\right\rangle := \frac{\sin\theta}{\sqrt{2}}\left(-\left|0\right\rangle + \left|1\right\rangle\right) + \frac{\cos\theta}{\sqrt{2}}\left(\left|0\right\rangle - \left|1\right\rangle\right).$$

Let $\left\{\left|\xi_i\right\rangle : i \in [4]\right\}$ be a set of orthonormal vectors.

By Lemma α is incommensurate with π, therefore, $\cup$ generates a dense subgroup of $H_1 := SO\left(span\left\{\left|\xi_3\right\rangle, \left|\xi 4\right\rangle\right\}\right)$. Note that $\Lambda\left(\sigma^x\right)[1,2]$ preserve $\left|\xi_1\right\rangle$, but not span $\left\{\left|\xi_2\right\rangle\right\}$. Therefore, by Lemma the set $H_1 \bigcup \Lambda\left(\sigma^x\right)[1,2] H_1 \Lambda\left(\sigma^x\right)[1,2]$ generates a dense subgroup of $SO\left(span\left\{\left|\xi_i\right\rangle : i = 2,3,4\right\}\right) =: H_2$, thus so does $\left\{\cup, \Lambda\left(\sigma^x\right)[1,2]\right\}$. Finally, observe that $\Lambda\left(\sigma^x\right)[2,1]$ does not preserve span $\left\{\left|\xi_1\right\rangle\right\}$, therefore, apply Lemma again we conclude that $\left\{\cup, \Lambda\left(\sigma^x\right)[1,2], \Lambda\left(\sigma^x\right)[2,1]\right\}$ generates a dense subgroup of $SO(4)$.

Proof of Theorem. Define $\cup := \left(H \otimes H \otimes H \cdot \Lambda^2\left(\sigma^x\right)[1,2,3]\right)^2$. Direct calculation shows that $\cup$ has eigenvalue 1 with multiplicity 6, and the other two eigenvalues $\lambda_\pm := \exp(\pm i\alpha)$, where $\alpha = \pi - \arccos\frac{3}{4}$. Since $\lambda_\pm$ are roots of the irreducible polynomial $\lambda^2 - \frac{3}{2}\lambda + 1$, which is not integral, therefore $\lambda_\pm$ are not algebraic integers. Thus α is incommensurate with π, which implies that $\cup$ generates a dense subgroups of the irritations over the corresponding eigenspace (denote the eigenvectors by $\left|\xi_7\right\rangle$ and $\left|\xi_8\right\rangle$). By direct calculation, the eigenvectors correspond to eigenvalue 1 are:

$$\left\{\left|000\right\rangle, \left|010\right\rangle, \left|100\right\rangle, \left|001\right\rangle + \left|011\right\rangle, \left|101\right\rangle + \left|110\right\rangle + \left|111\right\rangle, \left|011\right\rangle + \left|101\right\rangle\right\}.$$

Label the above eigenvectors by $\left|\xi_i\right\rangle, i \in [6]$. It is easy to verify that each $\cup_i, i \in [6]$, constructed below preserves $\left\{\left|\xi_j\right\rangle : 1 \le j < i\right\}$, but not span $\left\{\left|\xi_i\right\rangle\right\}$.

$\cup_1 := I \otimes I \otimes H,$	$\cup_2 := \cup_1 \cdot \Lambda^2\left(\sigma^x\right)[2,3,1] \cdot \cup_1,$
$\cup_3 := \cup_1 \cdot \Lambda^2\left(\sigma^x\right)[1,3,2] \cdot \cup_1,$	$\cup_4 := \Lambda^2\left(\sigma^x\right)[2,3,1],$
$\cup_5 := \cup_1 \cdot \Lambda^2\left(\sigma^x\right)[2,3,1] \cdot \cup_1,$	$\cup_6 := \Lambda^2\left(\sigma^x\right)[1,3,2].$

Applying Lemma several times, we see that $\left\{\cup, \cup_i, \cup_{i+1}, \cdots, \cup_6\right\}$ generates a dense subgroup of span $\left\{\left|\xi_j\right\rangle : i \le j \le 8\right\}$. Thus $\left\{\Lambda^2\left(\sigma^x\right) H\right\}$ generates a dense subgroup of SO(8). We leave the details for the interested leaders.

Example: Alternative proof for Theorem. Fix the arbitrary basis-changing real single-qubit gate S, and the basis $B := \left\{S, \Lambda^2\left(\sigma^x\right)\right\}$. We give an explicit construction to approximate an arbitrary real gate using the basis B. Due to the following result by Barenco et al., we need only consider approximating single-qubit real gates:

Proposition (Barenco et al.). Any gate on r qubits can be realized by $O\left(r^2 4^r\right)$ CNOT and single-qubit gates.

Fix any arbitrary single-qubit gate W that we would like to approximate. Without loss of generality, we can assume that S and W are rotations, for otherwise σ^x S and σ^x W are. For any $\beta \in [0, 2\pi)$, define

$$\left|\phi_\beta\right\rangle := \cos\beta\left|0\right\rangle + \sin\beta\left|1\right\rangle \quad \textit{and} \quad \cup_\beta := \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix}.$$

Let $\theta, \alpha \in [0, 2\pi)$, and θ not an integral multiple of $\pi/2$, be such that $\mathrm{S} \equiv \cup_\theta$ and $\mathrm{W} \equiv \cup_\alpha$. The following proposition can be easily checked.

Proposition. Let $\mathrm{W}_{\alpha/2}$ be a gate on $k+1$ qubits that $\mathrm{W}_{\alpha/2}\left|0\right\rangle^{\otimes k+1}=\left|\phi_{\alpha/2}\right\rangle\otimes\left|0\right\rangle^{\otimes k}$. With

$$W_\alpha := W_{\alpha/2}\left(-\overline{\Lambda}^{k+1}(-1)\right)W_{\alpha/2}^{\dagger}\sigma^z[1], \tag{4}$$

for any vector $\left|\xi\right\rangle\in H$,

$$\cup_\alpha\left|\xi\right\rangle\otimes\left|0\right\rangle^{\otimes k}=W_\alpha\left(\left|\xi\right\rangle\otimes\left|0\right\rangle^{\otimes k}\right) \tag{5}$$

Clearly $\overline{\Lambda}^{k+1}(-1)$ can be realized by $\Lambda^2(\sigma^x)$ and σ^z. Therefore, to approximate $\cup_\alpha$, it suffices to approximate σ^z and $\mathrm{W}_{\alpha/2}$, which we will show in the following subsections.

Define the constants $\delta_\theta := 1/\log\dfrac{1}{\cos^4\theta+\sin^4\theta}$, *and*, $\delta'_\theta := 1/\log\dfrac{1}{\cos^2\theta}$.

Approximating σ^z**.** If θ is a multiple of $\pi/4$, say $\theta=\pi/4$, then we can easily do a sign-flip by applying a bit-flip on $\cup_\theta\left|1\right\rangle=\frac{1}{-\sqrt{2}}\left|0\right\rangle+\frac{1}{\sqrt{2}}\left|1\right\rangle$. But for a general θ, $\cup_\theta\left|1\right\rangle=-\sin\theta\left|0\right\rangle+\cos\theta\left|1\right\rangle$ is "biased". Immediately comes into mind is the well-known idea of von Neumann on how to approximate a fair coin by tossing a sequence of coins of identical bias. That is, toss two coins, declare "0" if the outcomes are "01", declare "1" if the outcomes are "10", and continue the process otherwise. To illustrate the idea, consider

$$\cup_\theta\left|0\right\rangle\otimes\cup_\theta\left|1\right\rangle=\sin\theta\cos\theta\left(\left|11\right\rangle-\left|00\right\rangle\right)+\cos^2\theta\left|01\right\rangle-\sin^2\theta\left|10\right\rangle.$$

If we switch $\left|00\right\rangle$ and $\left|11\right\rangle$ and leave the other two base vectors unchanged, the first term on the right-hand side changes the sign, while the remaining two terms are unchanged. While we continue tossing pairs of "quantum coins" and do the $\left|00\right\rangle$-and-$\left|11\right\rangle$ switch, we approximate the sign-flip very quickly.

The state defined below will serve the role of $\frac{1}{\sqrt{2}}\left|1\right\rangle-\frac{1}{\sqrt{2}}\left|0\right\rangle$.

Definition. For any integer $k\geq 0$, the phase ancilla of size k is the state

$$\left|\Phi_k\right\rangle := \left(\cup_\theta\left|0\right\rangle\otimes\cup_\theta\left|1\right\rangle\right)^{\otimes k}.$$

Clearly $\left|\Phi_k\right\rangle$ can be prepared from $\left|0\right\rangle^{\otimes 2k}$ by B-circuit of size $O(k)$.

Lemma. The operator σ^z can be approximated with precision ε, for any $\varepsilon>0$, by a B-circuit of size $O(k)$, using the phase ancilla $\left|\Phi_k\right\rangle$, for some integer $k=O\left(\delta_\theta\log\frac{1}{\varepsilon}\right)$.

Proof. Let k be an integer to be determined later. The following algorithm is a description of a circuit approximating σ^z using $\left|\Phi_k\right\rangle$.

Algorithm 1

A B-circuit $\tilde{\sigma}^z$ approximating σ^z using the phase ancilla $\left|\Phi_k\right\rangle$.

Let $\left|b_0\right\rangle\otimes\left|b\right\rangle$ be a computational base vector, where $b_0\in\{0,1\}$ is the qubit to which σ^z is to applied, and $b=b_1b'_1b_2b'_2\cdots b_kb'_k\in\{0,1\}^{2k}$ are the ancilla qubits. Condition on b_0 (that is, if $b_0=0$, do nothing, otherwise do the following),

Case 1:	There is no I that $b_i\oplus b'_i$, do nothing.
Case 2:	Let I be the smallest index such that $b_i\oplus b'_i=0$, flip b_i and b'_i.

Clearly the above algorithm can be carried out by $O(k)$ applications of Toffoli. Fix an arbitrary unit vector $|\xi\rangle \in H$. Since neither σ^z nor $\tilde{\sigma}^z$ changes $|0\rangle\langle 0|(|\xi\rangle \otimes \langle \Phi_k|)$,

$$\left\| \sigma^z |\xi\rangle \otimes |\Phi_k\rangle - \tilde{\sigma}^z (|\xi\rangle \otimes |\Phi_k\rangle) \right\| \le \left\| -|1\rangle \otimes |\Phi_k\rangle - \tilde{\sigma}^z (|1\rangle \otimes |\Phi_k\rangle) \right\|. \qquad (6)$$

Let $|\Phi_k^+\rangle$ $(|\Phi_k^-\rangle)$ be the projection of $|\Phi_k\rangle$ to the subspace spanned by the base vectors satisfying Case (1) (Case (2)), it is easy to prove by induction that

$$\tilde{\sigma}^z (|1\rangle \otimes |\Phi_k^+\rangle) = |1\rangle \otimes |\Phi_k^+\rangle \ and \ \ \tilde{\sigma}^z (|1\rangle \otimes |\Phi_k^-\rangle) = -|1\rangle \otimes |\Phi_k\rangle.$$

Furthermore, $\left\| |\Phi_k^+\rangle \right\| = (\cos^4\theta + \sin^4\theta)^{k/2}$. Therefore, the left-hand side of Eq. (6) is upper bounded by $2\left\| |\Phi_k^+\rangle \right\| = 2(\cos^4\theta + \sin^4\theta)^{k/2}$. Since θ is not a multiple of $\pi/2$, the right-hand side is <1. Thus choosing $k = O(\delta_\theta \log \frac{1}{\varepsilon})$, the right-hand is the above can be made $\le \varepsilon$.

Creating $|\phi_{\alpha/2}\rangle$. We would like to construct a circuit that maps $|0\rangle \otimes |0\rangle^{\otimes k}$ to a state close to $|\phi_{\alpha/2}\rangle \otimes |0\rangle^{\otimes k}$. The main idea is to create a "logical" $|\phi_{\alpha/2}\rangle$:

$$|\hat{\phi}_{\alpha/2}\rangle := \cos\frac{\alpha}{2}|\hat{0}\rangle + \sin\frac{\alpha}{2}|\hat{1}\rangle, \qquad (7)$$

where $|\hat{0}\rangle$ and $|\hat{1}\rangle$ are two orthonormal vectors in a larger space spanned by ancillae, and the undo the encoding to come back to the computational basis. To create $|\hat{\phi}_{\alpha/2}\rangle$, we first create a state almost orthogonal to $|\hat{0}\rangle$, and then apply Grover's algorithm to rotate this state toward $|\hat{\phi}_{\alpha/2}\rangle$. Define the operator T_θ on 2 qubits as

$$T_\theta := \cup_{-\theta}[1]\Lambda(\sigma^x)[1,2]\cup_\theta[1]. \qquad (8)$$

Since for any $\beta, \cup_{-\beta} = \sigma^x \cup_\beta \sigma^x, T_\theta$ and $\Lambda(T_\theta)$ can be realized by the basis B.

Let $\theta_1 := \{\Lambda^2(\sigma^x), \sigma^z, \cup_\theta \cup_{-\theta}, T_\theta, \Lambda(T_\theta)\}$.

*Lemma***.** For any $\varepsilon > 0$ there exists a B_1-circuit $\tilde{W}_{\alpha/2}$ of size $O(\delta'_\theta \frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ that uses $O(\delta'_\theta \log\frac{1}{\varepsilon})$ ancilla and satisfies $\left\| \tilde{W}_{\alpha/2}|0\rangle^{\otimes k+1} - |\phi_{\alpha/2}\rangle \otimes |0\rangle^{\otimes k} \right\| \le \varepsilon.$

Proof. Let k > 0 be an integer to be specified later.

Define $|\hat{0}\rangle := |0\rangle^{\otimes 2k}$, $|\tilde{1}\rangle := T_\theta^{\otimes k}|\hat{0}\rangle$ *and* $\gamma := \arcsin(\cos^{2k}\theta)$. Notice that $\pi/2 - \lambda$ is the angle between $|\hat{0}\rangle$ and $|\tilde{1}\rangle$, and $0 < \gamma < \pi/2$, since $\sin\gamma = \langle\hat{0}|\tilde{1}\rangle$. Let S be the plane spanned by $|\hat{0}\rangle$ and $|\tilde{1}\rangle$. Let $|\hat{1}\rangle$ be the unit vector perpendicular to $|\hat{0}\rangle$ in S and the angle between $|\hat{1}\rangle$ and $|\tilde{1}\rangle$ is γ. Observe that on S we can do the reflection along $|\hat{1}\rangle$ and the reflection along $|\tilde{1}\rangle$. The former is simply $\overline{\Lambda}^{2k}(\sigma^z)$, which can be implemented using $\Lambda^2(\sigma^x)$ and σ^z. Since $\mathrm{T}_\theta^{-1} = T^\theta$, the reflection along $|\tilde{1}\rangle$ is $R := T_\theta^{\otimes k}(-\overline{\Lambda}^{2k}(\sigma^z))T_\theta^{\otimes k}$.

Without loss of generality we can assume $\alpha/2 < \pi/2$; otherwise we will rotate $|\tilde{1}\rangle$ close to $\overline{\Lambda}^{2k}(\sigma^z)|\hat{\phi}_{\alpha/2}\rangle$ and then apply $\overline{\Lambda}^{2k}(\sigma^x)$. Choose k sufficiently large so that $\gamma < \pi/2 - \alpha/2$. Now we can apply Grover's algorithm to rotate $\left|\tilde{1}\right\rangle$ to a state very close to $\left|\hat{\phi}_{\alpha/2}\right\rangle$. After that we do a "controlled-roll-back" to map $\left|\hat{1}\right\rangle$ (approximately) to $\left|1\right\rangle^{k}$ and does not change $\left|\hat{0}\right\rangle$. This will give us an approximation of $\left|\phi_{\alpha/2}\right\rangle$ in the state space of the controlling qubit. The algorithm is as follows. Let T be the integer such that $\left|\pi/2-(2T+1)\gamma-\alpha/2\right|<\gamma$. Then $T=O(1/\gamma)$.

Algorithm 2

A B_1-circuit $\tilde{W}_{\alpha/2}$ that maps $\left|0\right\rangle\otimes\left|0\right\rangle^{\otimes 2k}$ to a state close to $\left|\phi_{\alpha/2}\right\rangle\otimes\left|0\right\rangle^{\otimes 2k}$.

1.	Apply $\mathrm{I}\otimes T_\theta^{\otimes k}$.
2.	(Grover's algorithm) Apply $\left(R\hat{\Lambda}^{2k}\left(\sigma^z\right)\right)^T$.
3.	(Sub-circuit A_3) For a computational base vector $\left|b\right\rangle$ of the ancillae, if $\left|b\right\rangle\neq\left|\hat{0}\right\rangle$, flip the first bit.
4.	(Sub-circuit A_4) Use the first bit as the condition bit, apply $\Lambda\left(T_\theta^{\otimes k}\right)$.

It can be easily verified that $\left\|\tilde{W}_{\alpha/2}\left(\left|0\right\rangle\otimes\left|0\right\rangle^{\otimes 2k}\right)-\left|\phi_{\alpha/2}\right\rangle\otimes\left|0\right\rangle^{\otimes 2k}\right\|\leq 2\gamma$. Setting $\gamma\approx\varepsilon/2$, by direct computation the number of ancillae is $O(k)=O\left(\delta'_\theta\log\frac{1}{\varepsilon}\right)$, and the size of $\tilde{W}_{\alpha/2}$ is $O(k/\gamma)=O\left(\delta'_\theta\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)$.

Approximating $\bigcup_\alpha$. Theorems are a straightforward corollary of the following theorem and Proposition.

Theorem: For any $\varepsilon>0$, *the operator* $\bigcup_\alpha$ *can be approximated with precision* ε *by a B-circuit of size* $O\left(\delta_\theta\cdot\frac{1}{\varepsilon}\cdot\log\frac{1}{\varepsilon}\right)$ *and using* $O\left(\delta_\theta\cdot\log\frac{1}{\varepsilon}\right)$ *ancillae.*

Proof. We first compose a B-circuit that approximates $\bigcup_\alpha$, according Algorithm 2, and use k_1 (different) ancillae in each call to the latter, for an integer k_1 to be specified later. Let $\gamma:=\cos^{2k_1}\theta$. Then the precision is $O(\gamma)$. . After implementing T_θ and $\Lambda(T_\theta)$, there are in total $O\left(\frac{1}{\varepsilon}\right)$ uses of σ^z.

Finally we apply Algorithm 1 to approximate each σ^z using the same phase ancilla $\left|\phi_{k_2}\right\rangle$ for $k_2=O\left(1/\gamma^3\right)$. Let $\delta_\theta:=2\left(\cos^4\theta+\sin^4\theta\right)^{k_2/2}$ be the error of one call to $\tilde{\sigma}^z$ using exactly $\left|\phi_{k_2}.\right\rangle$ Observe that using the same phase ancilla for $O\left(\frac{1}{\gamma}\right)$ times causes error at most $1+2+\cdots+O\left(\frac{1}{\gamma}\right)-1=O\left(\frac{1}{\gamma^2}\right)$. Setting $\delta_\theta=\gamma^3$, the total error caused by $\tilde{\sigma}^z$ is $O(\gamma)$. Thus the total error of the whole circuit is still $O(\gamma)$. Setting $\gamma\approx\varepsilon, k_1=O\left(\delta'_\theta\log\frac{1}{\varepsilon}\right)=O\left(\delta_\theta\log\frac{1}{\varepsilon}\right)$ and $k_2=O\left(\delta_\theta\log\frac{1}{\varepsilon}\right)$. .

Therefore the number of ancilla is $O(k_1+k_2)=O\left(\delta_\theta\log\frac{1}{\varepsilon}\right)$. The size of the circuit is

$$O\left((k_1+k_2)\tfrac{1}{\varepsilon}\right)=O\left(\delta_\theta\tfrac{1}{\varepsilon}\log\tfrac{1}{\varepsilon}\right).$$

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, England. – 2000.
3. Hirvensalo M. Quantum computing. – Natural Computing Series, Springer-Verlag, Berlin. – 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. Lo H.-K., Popescu S. and Spiller T. (Eds). Introduction to quantum computing and information. – World Scientific Publ. Co. – 1998.
9. Berman G.P., Doolen G.D., Mainieri R. and Tsifrinovich V.I. Introduction to quantum computers. – World Scientific Publ. Co. – 1999.
10. Rieffel E. and Polak W. An introduction to quantum computing for non-physicists // ACM Computing Surveys. – 2000. – Vol. 32. – No 3. – pp. 300 – 335.
11. Hogg T., Mochon C., Polak W. and Rieffel E. Tools for quantum algorithms // International Journal of Modern Physics. – 1999. – Vol. C10. – No 7. – pp. 1347 – 1361.
12. Uesaka Y. Mathematical principle of quantum computation (in Japanese). – Corona Publ. Co. Ltd. – 2000.
13. Marinescu D.C. and Marinescu G.M. Approaching quantum computing. – Pearson Prentice Hall, New Jersey. – 2005.
14. Benenti G., Casati G. and Strini G. Principles of quantum computation and information. –Singapore: World Scientific. – Vol. I. – 2004; – Vol. II. – 2007.
15. Nakahara M. and Ohmi T. Quantum computing: From Linear Algebra to Physical Realizations. – Taylor & Francis. – 2008.
16. Stenholm S. and Suominen K.-A. Quantum approach to informatics. – Wiley- Interscience. A J. Wiley&Sons, Inc. – 2005.
17. Jaeger G. Quantum Information: An Overview. – N.Y.: Springer Verlag. – 2007.
18. McMahon D.Quantum computing explained. – Wiley- Interscience. A J. Wiley&Sons, Inc. – 2008.