Universal quantum computation with unlabelled qubits

2006 J. Phys. A: Math. Gen. 39 8507

(http://iopscience.iop.org/0305-4470/39/26/016)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **39** (2006) 8507–8513 doi:10.1088/0305-4470/39/26/016

Universal quantum computation with unlabelled qubits

Simone Severini

Department of Mathematics and Department of Computer Science, University of York, Heslington, YO10 5DD York, UK

E-mail: ss54@york.ac.uk

Received 19 January 2006
Published 14 June 2006
Online at stacks.iop.org/JPhysA/39/8507

Abstract

We show that an nth root of the Walsh–Hadamard transform (obtained from the Hadamard gate and a cyclic permutation of the qubits), together with two diagonal matrices, namely a local qubit-flip (for a fixed but arbitrary qubit) and a non-local phase-flip (for a fixed but arbitrary coefficient), can do universal quantum computation on n qubits. A quantum computation, making use of n qubits and based on these operations, is then a word of variable length, but whose letters are always taken from an alphabet of cardinality three. Therefore, in contrast with other universal sets, no choice of qubit lines is needed for the application of the operations described here. A quantum algorithm based on this set can be interpreted as a discrete diffusion of a quantum particle on a de Bruijn graph, *corrected on-the-fly* by auxiliary modifications of the phases associated with the arcs.

PACS number: 03.67.−a
Mathematics Subject Classification: 81P68

1. Introduction

The study of universality in quantum computation goes back to [3]. In the circuit model, universality has been considered by a number of papers (see [1, 9] and the references therein). Probably, the simplest universal set of gates consists of the Hadamard gate H together with the Toffoli gate T [9]. Of course, in order to have universality, we need the freedom of applying H and T to arbitrary qubits of the computer: H to any qubit and T to any three qubits. In fact, out of measurement processes, every computational step is induced by a unitary, obtained by tensoring together H's, T's and identity matrices. The number of different unitaries obtained in this way is then a function of the number of qubits.

In this paper, we define a universal set of unitary quantum operations depending on the computational space of the machine. This means that the unitaries change whenever the total

number of qubits changes. Since the operations act globally on all qubits, they do not require the choice of qubit lines at each computational step. This is in contrast to standard finite universal sets (e.g., H and T) where the gates remain fixed but may be applied to varying choices of qubit lines. The set defined here is composed by an nth root of the Walsh–Hadamard transform (constructed from the Hadamard gate and a cyclic permutation of the qubits), and two diagonal matrices: a (local) qubit-flip (for a fixed but arbitrary qubit) and a (non-local) phase-flip (for a fixed but arbitrary coefficient). To prove that the defined set is universal on n qubits, we reduce it to H and T, acting on any qubit and any three qubits. The characteristic property of the set is that the choice of the qubits to which apply H and T is not reflected into the structures of the unitaries. This observation would like to be a justification to the title of the paper. Of course, the set described here is inconvenient from the physical implementation point of view, because it requires nonlocal interaction between qubits (which is physically expensive). However, it may be useful to remark that the set provides a mathematical framework, in which doing universal quantum computation is constructing words whose letters are unitaries (two of which commute) taken from an alphabet of cardinality three. So, the outcome of a computation depends on the length of the word and the order of the letters. This has some flavour that reminds of quantum finite state automata and other sequential machines (see e.g. [4, 7]).

The remainder of this paper is organized as follows. In section 2, we give some preliminary definitions. In section 3, we formally state and prove the main result. This is done by proving universality with a reduction to $\{H, T\}$. It is not difficult to verify that the nth root of the Walsh–Hadamard transform respects the topology of the de Bruijn graph. Namely, the ijth entry of this unitary is nonzero if and only if there is a directed edge from the vertex labelled i to the vertex labelled j, in the de Bruijn graph on 2^n vertices. In section 4, in virtue of this observation, we point out that any quantum algorithm can be seen as the discrete diffusion of a quantum particle on a de Bruijn graph, *corrected on-the-fly* by a qubit-flip and a phase-flip, both fixed but arbitrary. This reminds of the context of discrete quantum walks [2] or the processes studied in [6]. A natural open question would be to prove that the nth root of the Walsh–Hadamard transform and a phase-flip (for a fixed but arbitrary coefficient) form a universal set.

2. Definitions

In this section we introduce some preliminary definitions.

Definition of V_n. We denote by V_n a square matrix of dimension 2^n such that $[V_n]_{i,j} \in \left\{0, \pm\frac{1}{\sqrt{2}}\right\}$ and with exactly the following nonzero entries:

$$\begin{aligned}
&[V_n]_{1,1} = [V_n]_{1,2^{n-1}+1} = [V_n]_{2,1} = \tfrac{1}{\sqrt{2}},\\
&[V_n]_{2,2^{n-1}+1} = -\tfrac{1}{\sqrt{2}},\\
&[V_n]_{3,2} = [V_n]_{3,2^{n-1}+2} = [V_n]_{4,2} = \tfrac{1}{\sqrt{2}},\\
&[V_n]_{4,2^{n-1}+2} = -\tfrac{1}{\sqrt{2}},\\
&\vdots\\
&[V_n]_{2^n-1,2^{n-1}} = [V_n]_{2^n-1,2^n} = [V_n]_{2^n,2^{n-1}} = \tfrac{1}{\sqrt{2}},\\
&[V_n]_{2^n,2^n} = -\tfrac{1}{\sqrt{2}}.
\end{aligned} \tag{1}$$

The matrix V_n is real-orthogonal and it is an nth root of $H_n := H^{\otimes n}$, where H is the 1-qubit Hadamard gate:

$$H := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2}$$

For example,

$$V_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix} \tag{3}$$

and $V_2^2 = H$. An alternative and more direct definition of V_n can be given as follows. Let S_n be the full symmetric group on the set $\{1, 2, \ldots, n\}$. We denote permutations of length n as ordered sets. For example, the elements of S_3 are $(1, 2, 3)$, $(1, 3, 2)$, $(3, 2, 1)$, $(2, 3, 1)$, $(2, 1, 3)$ and $(3, 1, 2)$. With an abuse of our notation, their regular permutation representations are denoted in the same way. The matrix V_n is defined as

$$V_n := P \cdot (H \otimes I^{\otimes n-1}), \tag{4}$$

where

$$P = (1, 3, \ldots, 2^n - 1, 2, 4, \ldots, 2^n). \tag{5}$$

This permutation is nothing but a cyclic-shift of the qubits:

$$P : |a_1 a_2 \ldots a_n\rangle \longrightarrow |a_{n-1} a_1 \ldots a_{n-2}\rangle. \tag{6}$$

This explains why $V_n^n = H_n$ ($H_n := H^{\otimes n}$). It is easy to verify that equations (1) and (4) define the same matrices. Notice that, when $n = 2$, the permutation P is the Swap-gate:

$$(1, 3, 2, 4) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{7}$$

Definition of $P_n(k)$. Let us denote by $P_n(k)$ the $2^n \times 2^n$ matrix defined by

$$[P_n(k)]_{i,j} := \begin{cases} 0 & \text{if} \quad i \neq j; \\ -1 & \text{if} \quad i = k \quad \text{with} \quad k \in \{1, \ldots, 2^n\}; \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

The matrix $P_n(k)$ is the Pauli operator

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

on the nth qubit controlled by all other qubits $1, 2, \ldots, n-1$ and then conjugated by the permutation that interchanges dimensions k and 2^n. For example,

$$P_2(3) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = I_2 \oplus XZX, \tag{9}$$

where X is the Pauli operator

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Definition of $F_n(k)$. Let us denote by $F_n(k)$ the $2^n \times 2^n$ matrix defined by

$$[F_n(k)]_{i,j} := \begin{cases} 0 & \text{if } i \neq j; \\ -1 & \text{if the } k\text{th qubit is1;} \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

The matrix $F_n(k)$ is the Pauli operator Z acting on the kth qubit and the identity on all other qubits. For example,

$$F_2(1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} = I \otimes Z \qquad \text{and} \qquad F_2(2) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} = Z \otimes I. \tag{11}$$

3. Main result

In this section we prove the following theorem.

Theorem 1. *The set*

$$B = \{V_n, P_n(i), F_n(j)\}, \tag{12}$$

for fixed but arbitrary $i \in \{1, \ldots, 2^n\}$ *and* $j \in \{1, \ldots, n\}$, *is universal for quantum computation on n qubits.*

By this theorem, any quantum computation in the circuit model can be seen as a word whose letters are taken from the alphabet B.

The *Toffoli gate* T is the three-qubit gate defined as $T : (a, b, c) \longrightarrow (a, b, ab \oplus c)$, where $\oplus$ denotes addition modulo 2 and $a, b, c \in \{0, 1\}$. The matrices defined by the Toffoli gate are then a special kind of transposition. If we label the states of the computational basis in lexicographic order, the matrices defined by the Toffoli gate on 3 qubits are

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \tag{13}$$

The set $B' = \{H, T\}$ is universal for quantum computation [1, 9]. We prove theorem 1 by showing that the set B is equivalent to the set B', once the number of qubits has been fixed.

In order to verify this equivalence, we show that any expression made by the tensor product of H, T and the 2×2 identity matrix I_2 corresponds to a sequence of the three elements of B defined in the statement of the above theorem. We will consider the set

$$D_n := \{P_n(i) : 1 \leqslant i \leqslant 2^n\}. \tag{14}$$

Lemma 2. *The set $A = V_n \cup D_n$ is universal for quantum computation on n qubits.*

Proof. By the definition of V_n (equation (4) above), we have

$$P^{-1} \cdot V_n = H \otimes I^{\otimes n-1}, \tag{15}$$

where

$$P^{-1} = (1, 2^{n-1}+1, 2, 2^{n-1}+2, 3, \ldots, 2^{n-1}, 2^n) = (1, 3, \ldots, 2^n - 1, 2, 4, \ldots, 2^n)^{-1}.$$

This means that if we can construct the matrix P^{-1} then we can also construct the Hadamard gate. First, observe that

$$V_n^{-1} = V_n^{2n-1} = V_n^T,$$

since

$$V_n^{2n} = V_n^n V_n^n = H_n H_n = I^{\otimes n}.$$

The matrix V_n^T can be then obtained directly from V_n. The nonzero entries in the last two rows of V_n are

$$[V_n]_{2^n-1,2^{n-1}} = [V_n]_{2^n-1,2^n} = [V_n]_{2^n,2^{n-1}} = \tfrac{1}{\sqrt{2}},$$
$$[V_n]_{2^n,2^n} = -\tfrac{1}{\sqrt{2}}.$$

The nonzero entries in the last two columns of $Q_n = P_n(2^n) \cdot V_n^T$ are

$$[Q_n]_{2^{n-1},2^n-1} = [Q_n]_{2^n,2^n-1} = \left[P_n(2^n) \cdot V_n^T\right]_{2^{n-1},2^n} = \tfrac{1}{\sqrt{2}},$$
$$[Q_n]_{2^n,2^n} = -\tfrac{1}{\sqrt{2}}.$$

Since

$$H \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = X,$$

it follows that

$$V_n Q_n = V_n P_n(2^n) V_n^T = I_{2^n-2} \oplus X = (1, 2, \ldots, 2^n - 2, 2^n, 2^n - 1), \tag{16}$$

which is indeed the Toffoli gate. Now, given that

$$S_{2^n} = \langle (1, 2, \ldots, 2^n - 2, 2^n, 2^n - 1), (2, 3, \ldots, 2^n, 1) \rangle,$$

if we can construct the cyclic permutation $(2, 3, \ldots, 2^n, 1)$ then we will have the full symmetric group S_{2^n}. If we can construct S_{2^n} then we will get P^{-1} and all the permutations that we need for applying H and T to arbitrary qubits. The permutation $(2, 3, \ldots, 2^n, 1)$ can be constructed with the following procedure (most probably not optimal):

(i) Let $M_1 = H_n F_n(n) H_n = X \otimes I^{\otimes n-1}$. By applying elements from D_n, transform the bottom-left block of M_1 into the diagonal matrix $dia(p_1, \ldots, p_{2^{n-3}}, -p_{2^{n-3}+1}, \ldots, -p_{2^{n-2}})$, where $p_i = (1, -1)$. Let M_2' be the matrix obtained in this way.

(ii) Let $M_2 = V_n^{-1} M_2' V_n$. By applying elements from D_n, transform the top-right block of M_2 into the identity matrix and the bottom-left block into the diagonal matrix $\mathrm{dia}(p_1, \ldots, p_{2^{n-4}}, -p_{2^{n-4}+1}, \ldots, -p_{2^{n-3}})$.

(iii) Repeating the second step $n-2$ times one gets the permutation matrix $(2, 3, \ldots, 2^n, 1)$. (This can be easily checked with any computer algebra system.)

An example with three qubits may help to clarify the procedure:

$$H_3(1, 2, 3, 4, \overline{5}, \overline{6}, \overline{7}, \overline{8})H_3 = (5, 6, 7, 8, 1, 2, 3, 4),$$
$$V_3^{-1}(5, 6, 7, 8, 1, \overline{2}, \overline{3}, 4)V_3 = (3, 4, 5, \overline{6}, 7, 8, 1, \overline{2}),$$
$$V_3^{-1}(3, 4, 5, 6, 7, 8, 1, \overline{2})V_3 = (2, 3, 4, 5, 6, 7, 8, 1).$$

The notation is easily explained:

$$\left(5, 6, 7, 8, 1, \overline{2}, \overline{3}, 4\right) = \left(\begin{array}{cccc|cccc} & & & & 1 & 0 & 0 & 0 \\ & & & & 0 & 1 & 0 & 0 \\ & \mathbf{0} & & & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & & & & \\ 0 & -1 & 0 & 0 & & & & \\ 0 & 0 & -1 & 0 & & \mathbf{0} & & \\ 0 & 0 & 0 & 1 & & & & \end{array}\right).$$

By the above constructions, we have the following fact: the matrix group $G = \langle V_n, D_n \rangle$ has the regular permutation representation of S_{2^n} as a subgroup. Then T, obtained with equation (16), can be applied to any three qubits; H, obtained from equation (15), can be applied to any qubit. Since $B' = \{H, T\}$ is universal, the lemma follows. □

Lemma 3. *The matrix group $G = \langle V_n, P_n(i), F_n(j) \rangle$, for fixed but arbitrary $i \in \{1, \ldots, 2^n\}$ and $j \in \{1, \ldots, n\}$, contains the set D_n.*

Proof. By the definitions,

$$\begin{aligned} V_n^k F_n(n) V_n^{-k} &= (P(H \otimes I^{\otimes n-1}))^k \cdot (Z \otimes I^{\otimes n-1}) \cdot (P(H \otimes I^{\otimes n-1}))^{-k} \\ &= P^k (H \otimes I^{\otimes n-1}) \cdot (Z \otimes I^{\otimes n-1}) \cdot (H \otimes I^{\otimes n-1}) P^{-k} \\ &= P^k (X \otimes I^{\otimes n-1}) P^{-k} \\ &= I \otimes \cdots \otimes I \otimes X \otimes I \cdots \otimes I, \end{aligned}$$

where X is at the kth position of the tensor product. For example, $V_3^1 F_3(3) V_3^{-1} = I \otimes I \otimes X$. Let H be the matrix group generated by the matrices $I \otimes \cdots \otimes I \otimes X$, $I \otimes \cdots \otimes I \otimes X \otimes I$, $X \otimes I \otimes \cdots \otimes I$. The group H is isomorphic to $\mathbb{Z}_2^n$ (indeed, the above matrices are the permutation representations of the standard generators of $\mathbb{Z}_2^n$). Since, for every $i, j \in \{1, \ldots, 2^n\}$, there is an element of H sending $P_n(i)$ to $P_n(j)$, we can construct D_n, that is the set of all $2^n \times 2^n$ diagonal matrices with entries 1 and -1. We considered $F_n(n)$, but we could also take $F_n(j)$, for any $j \in \{1, \ldots, n\}$, without loss of generality. □

Theorem 1 is a consequence of lemma 2 together with lemma 3.

4. A relation with de Bruijn graphs

Let Σ be an alphabet of cardinality d and let Σ_n^* be the set of all words of length n over Σ. The *d-ary n-dimensional de Bruijn graph* is a directed graph denoted by $B(d, n)$ and defined as follows [5]: the set of vertices is Σ_k^*; there is an arc from i to j if and only if the last $n-1$ letters of i are the same as the first $n-1$ letters of j. The graph $B(2, n)$ is called (*directed*) *binary*

de Bruijn graph. Note that $B(2, n)$ has exactly two loops: one loop is at the vertex $0 \ldots 0$; the other one at the vertex $1 \ldots 1$. These graphs have important applications in cryptography and distributed computing. In particular, they provide some of the best-known topologies for communication networks. For example, the Galileo space probe of NASA used a network based on a de Bruijn graph to implement a signal decoder [8]. Let M_n be the adjacency matrix of $B(2, n)$. The rows and the columns of this matrix can be ordered in such a way that

$$\begin{aligned}
&[M_n]_{1,1} = [M_n]_{1,2^{n-1}+1} = [M_n]_{2,1} = [M_n]_{2,2^{n-1}+1} = 1,\\
&[M_n]_{3,2} = [M_n]_{3,2^{n-1}+2} = [M_n]_{4,2} = [M_n]_{4,2^{n-1}+2} = 1,\\
&\vdots\\
&[M_n]_{2^n-1,\frac{1}{2}2^n-1} = [M_n]_{2^n-1,2^n} = [M_n]_{2^n,\frac{1}{2}2^n-1} = [M_n]_{2^n,2^n} = 1.
\end{aligned}$$

Tanner [11] pointed out that the matrix V_n is obtained from the matrix M_n by negating the entries $[M_n]_{2,2^{n-1}+1}, [M_n]_{4,2^{n-1}+2}, \ldots, [M_n]_{2^n,2^n}$ and rescaling M_n by $\frac{1}{\sqrt{2}}$. It is easy to see that a simple random walk on $B(d, n)$ converges very quickly to uniformity, in fact it is perfectly mixed after n steps. *Simple* means that at each vertex the walker chooses to cross an incident edge by tossing a fair die with d faces. We associate the vertices of $B(2, n)$ with the elements of the computational basis $|0\rangle \equiv |0 \ldots 0\rangle, |1\rangle \equiv |0 \ldots 01\rangle, \ldots, |2^{n-1}\rangle \equiv |1 \ldots 1\rangle$. For every $|i\rangle$ and $|j\rangle$, there is a diagonal matrix $Q \in D_n$ such that $H_n Q H_n |i\rangle = V_n^n Q V_n^n |i\rangle = |j\rangle$. We may interpret this process as a discrete quantum walk on $B(2, n)$ induced by V_n and *corrected on-the-fly* by an appropriate diagonal unitary with ± 1 entries. If the walk is not corrected then $V_n^{2n}|i\rangle = |i\rangle$ for every i, given that H_n is symmetric. The walk on $B(2, n)$ is perfectly mixed at the nth step, but it can be driven with probability 1 to any vertex in exactly $2n$ steps. For example, the following is an algorithm that takes the state $|0\rangle$ to the state $|2^n - 1\rangle$ in $2n$ steps:

$$V_n^n|0\rangle = |+\rangle^{\otimes n}, \qquad F_1(1)^{\otimes n}|+\rangle^{\otimes n} = |\psi\rangle, \qquad V_n^n|\psi\rangle = |2^n - 1\rangle.$$

(A curiosity: the positions of the minus sign in the state $|\psi\rangle$ correspond to the numbers with an odd number of 1's in their binary expansion (A007413 [10]).) A generalization for any $|i\rangle$ and $|j\rangle$ is straightforward.

Acknowledgments

I would like to thank the anonymous referees. Their comments helped me to improve the presentation of this paper.

References

[1] Aharonov D 2003 Simple proof that Toffoli and Hadamard are quantum universal *Preprint* quant-ph/0301040
[2] Ambainis A 2004 Quantum walks and their algorithmic applications *Preprint* quant-ph/0403120
[3] Deutsch D 1989 Quantum computational networks *Proc. R. Soc.* A **425** 73–90
[4] Gudder S 1999 Quantum automata: an overview *Int. J. Theor. Phys.* **38** 2261–82
[5] Heydemann M-C 1997 Cayley graphs and interconnection networks *Graph Symmetry (Montreal, PQ, 1996)* (*NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.* vol 497) (Dordrecht: Kluwer) pp 167–224
[6] Košík J 2005 Scattering quantum random walk *Opt. Spectrosc.* **99** 224–6
[7] Moore C and Crutchfield J P 2000 Quantum automata and quantum grammars *Theor. Comput. Sci.* **237** 275–306
[8] Mukherjee B 1997 *Optical Communication Networks* (*Series on Computer Communications*) (New York: McGraw-Hill)
[9] Shi Y 2002 Both Toffoli and controlled-Not need little help to do universal quantum computation *Preprint* quant-ph/0205115
[10] Sloane N J A 2005 The on-line encyclopedia of integer sequences, www.research.att.com/njas/sequences/
[11] Tanner G 2000 Spectral statistics for unitary transfer matrices of binary graphs *J. Phys. A: Math. Gen.* **33** 3567–85