Quantum Computing

Proseminar in Theoretical Physics
Institut für theoretische Physik
ETH Zürich

Prof. Dr. Helmut G. Katzgraber
Prof. Dr. Renato Renner

SS07

typeset, compilation & editing by H. G. Katzgraber

Table of Contents

TABLE OF CONTENTS

Chapter 1

Computational Models for Quantum Computation

Pascal Stephan Philipp Steger
supervisor: Dr. Stefan Hohenegger

The quantum Turing machine and quantum networks are described as the two basic models of quantum computation. The quantum mechanical descriptions using S-matrices and Hamiltonians for the quantum gates as well as the step operator T for both gates and the quantum Turing machine are summarized. Finally, connections between the two models are established through theorems from complexity theory.

1.1 Introduction

This article will give you an overview over the quantum analoga to the previously shown classical computation models, the Turing machine and the circuit model. There exist two basic models for quantum computation: The quantum computational network or circuit, built of quantum gates and the Quantum Turing Machine (hereafter QTM). The differences between those concepts will be highlighted. Connections between the two models are established through complexity theory and the step operator, which is a quantum mechanical description. The differences between classical and quantum computation will be highlighted wherever possible. Along the way some mathematical framework and examples of gates are introduced.
I will concentrate on the articles by Deutsch ([1], [2]) for quantum networks and a proof for a universal quantum gate, ¡rticles by Benioff ([3]) for quantum Turing machines and Yao ([4]) for complexity theory.

1.2 Quantum Gates

1.2.1 Logic Circuits

The well-known classical computer performs computations with logic circuits. Let us define what exactly this means: A *computation* is a process that produces output depending on some input. *Input* and *output* denote abstract symbols. A *bit* or *quantum* is the smallest possible quantity of non-probabilistic information. The information is physically represented in a carrier, e.g. a transistor, or a spin 1/2-particle, which is more useful in the context of quantum computation.

A *logic circuit* is a computing machine consisting of logic gates. The computational steps are synchronized. The outputs of step (i) can be used as inputs for step $(i+1)$. The symbols in the example circuit - fig. 1.1 - stand for different parts of a general circuit: Gates are denoted by rectangles with input connections to the left and output connections to the right. Sources and sinks, denoted by triangles and crosses, can be used to implement special input and output conditions: A *source* is a gate with only one output that emits 0 or 1 in each step, it is reversible. A sink on the other hand has only one input and deletes information; it is irreversible. A *unit wire* computes the identity function, a fixed time dilation is indicated as number in a circle.

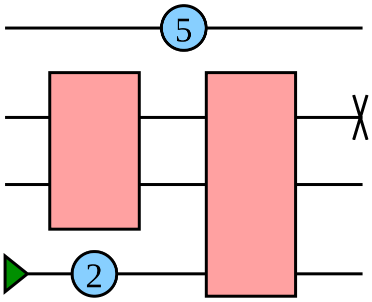

Figure 1.1: Example for a logic circuit showing different pieces.

Physical processes in a quantum computer, and more specifically in a gate, follow three steps:

1. preparation of the input states in carriers
2. QM elastic scattering (errorless)
3. measurement of output carriers after a fixed number of steps

The second process can be seen as happening in a black box. The actual scattering and projection of the state are implementation-specific and do not interfere with

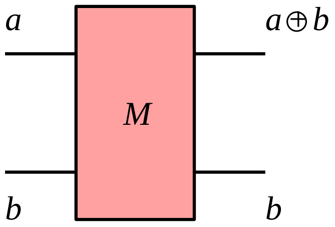

Figure 1.2: measurement gate - XOR - CNOT

the theoretical model.
What is a *logic gate*? It is a computing machine, where input and output consist of fixed number of bits. Some fixed computation is done in fixed time. A *quantum gate* can have quantum mixtures of eigenstates of the input observable $\hat{I}$ and output observable $\hat{O}$ as input/output. These are called qubits. A *reversible gate* has the property that inputs and outputs are related by an invertible function – in the ideal case, if no errors occur. No information is deleted, therefore the energy loss of $k_B T \ln 2$ - called Landauerś principle - does not occur. Any irreversible gate can be converted to a reversible one by copying an appropriate number of inputs to the output. Reversible gates and circuits have the same number of input and output wires.

1.2.2 Mathematical Descriptions

Several mathematical descriptions of gates are possible. Tables, permutation and the S-matrix are explained.

Tables

We choose a *computational basis* by the eigenstates of input operator $\hat{I}$ and output operator $\hat{O}$ in the Schrödinger picture. They both have to be the same for practical reasons. In this basis one can write down the action of a gate using a table. The following example corresponds to the gate in fig. 1.2.

a	b	$a \oplus b$	b	$(a \oplus b) \oplus b$	b
0	0	0	0	0	0
0	1	1	1	0	1
1	0	1	0	1	0
1	1	0	1	1	1

(1.1)

This gate is called XOR since it computes the logical XOR function from inputs a and b, copying a as a second output to guarantee reversiblity. Its action can also be understood as inverting the b input if a is set and returning b unchanged otherwise. Therefore it is also referred to as CNOT – controlled NOT – or measurement gate.

Permutations

Another way of description uses permutations: let $\{|a,b\rangle\}, \quad a,b \in \{0,1\}$ be the four computational basis states for a system with two inputs and outputs. A gate transforms inputs into outputs, for our example

$$\begin{array}{ccc} |0,0\rangle & \rightarrow & |0,0\rangle \\ |0,1\rangle & \rightarrow & |1,1\rangle \\ |1,0\rangle & \rightarrow & |1,0\rangle \\ |1,1\rangle & \rightarrow & |0,1\rangle \end{array} \tag{1.2}$$

S-matrix

A third way is given by using a S-Matrix, this is most suitable for quantum gates. In principle, it connects the two previous descriptions using linear algebra. A unitary matrix $S^{ab\dots}_{a'b'\dots}$ has clumped indices $ab\dots$, $a'b'\dots$ denoting the states of the input and output carriers. The operation of a gate corresponds to a matrix multiplication with $S^{ab\dots}_{a'b'\dots}$,

$$|a,b,\dots\rangle \rightarrow \sum_{a',b',\dots\in\{0,1\}} S^{ab\dots}_{a'b'\dots}|a',b',\dots\rangle \equiv S|a,b,\dots\rangle. \tag{1.3}$$

Unitarity of S is necessary to describe a reversible gate. In this picture repeated gates are represented by powers of S. If no basis is chosen explicitly, S can also denote a general linear operator.

1.2.3 Examples

NOT

For an example of the action of the S-matrix, consider the NOT gate in fig. 1.3. Its S-matrix is given by

$$S_N = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{1.4}$$

as one can check by multiplication with $(1,0)$ or $(0,1)$, which are vectors denoting the states $|1\rangle$ and $|0\rangle$ in the basis $\{|0\rangle,|1\rangle\}$. Powers of S correspond to several copies of NOT after each other, $\alpha \in \mathbb{N}$ implies that N^α is a logic gate: identity

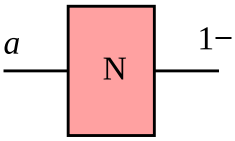

Figure 1.3: NOT gate

or NOT. But one can also take a power $\alpha \notin \mathbb{N}$ of the operator N: N^α does then not describe the action of a logic gate anymore, but the action of a more general quantum gate.

$$S_{N^\alpha} = S_N^\alpha = \frac{1}{2}\begin{pmatrix} 1+e^{i\pi\alpha} & 1-e^{i\pi\alpha} \\ 1-e^{i\pi\alpha} & 1+e^{i\pi\alpha} \end{pmatrix} \tag{1.5}$$

Toffoli and Q

The Toffoli gate from classical computation has an analogon in quantum computation, it is depicted in fig. 1.4. Its classical version can be described by the S-matrix

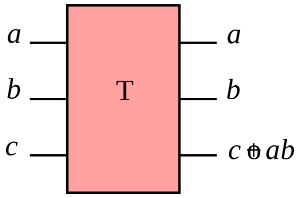

Figure 1.4: Toffoli gate

$$S^{abc}_{Ta'b'c'} = \delta^a_{a'}\delta^b_{b'}[(1-ab)\delta^c_{c'} + ab(S_N)^c_{c'}] \tag{1.6}$$

Analogously, the quantum gate Q reads as

$$S^{abc}_{Qa'b'c'} = \delta^a_{a'}\delta^b_{b'}\left[(1-ab)\delta^c_{c'} + iabe^{-i\pi\alpha/2}(S_N^\alpha)^c_{c'}\right] \tag{1.7}$$

and boils down to the classical form if $\alpha \in \mathbb{N}$.

If we want to explicitly write these two gates down, we have to use a certain basis, e.g. $0 = |000\rangle$, $1 = |001\rangle$, …, $6 = |110\rangle$, $7 = |111\rangle$. The S-matrices are then given by

$$\begin{aligned} S_T &= \begin{pmatrix} \mathbb{1}_6 & & \\ & 0 & 1 \\ & 1 & 0 \end{pmatrix} \\ S_Q &= \begin{pmatrix} \mathbb{1}_6 & & \\ & i\cos\pi\alpha/2 & \sin\pi\alpha/2 \\ & \sin\pi\alpha/2 & i\cos\pi\alpha/2 \end{pmatrix} \end{aligned}$$

1.2.4 Equivalence

There is a problem with using quantum gates and performing logical operations with them: Consider for example the repeated use of a NOT-gate,

$$\begin{aligned} S_{N^2} &= S_N^2 = \mathbb{1}, && (1.8)\\ (S_{N^\alpha})^m &= S_N^{m\alpha} = S_N^{m\alpha - 2\lfloor m\alpha/2 \rfloor}. && (1.9) \end{aligned}$$

The exponent $1+\varepsilon$ is arbitrarily close to 1, but never exact for an irrational α, $m \in \mathbb{N}$, a failure being possible. The time before non-classical behaviour is given by the reciprocal of the expectation value for the wrong result,

$$t = \frac{1}{\max_{|\Psi\rangle}(1 - |\langle\Psi|S_N^\dagger S_{N^\alpha}^m|\Psi\rangle|^2)} = \frac{1}{\sin^2 \pi\varepsilon/2} \sim \varepsilon^{-2} \overset{(\varepsilon\to 0)}{\longrightarrow} \infty.$$

Two circuits are called *computationally equivalent*, if they yield the same output given the same input. Exact equivalence is not possible in QM[1]. One needs to introduce another notion: F and G are *adequate sets*, if there exists a series $\{g_n \in G\}$ for all $f \in F$ and a sequence $\{\phi_n\}$ of phase angles such that

$$\lim_{n\to\infty} S_{g_n} e^{i\phi_n} = S_f \qquad (1.10)$$

As an example, $F = \{N\}$ and $G = \{N^\alpha, \mathbb{1}\}$ are adequate. One now wants to find a *universal gate*, that is a quantum gate such that the set of unit wire, source and this gate is adequate to set of all possible gates. This is exactly what the Toffoli gate is for classical gates.

Claim:

The Q-gate is universal to the set of all quantum gates.

Proof:

Create repetoire of gates that Q is adequate to:

1. Toffoli gate
2. all logic gates
3. all 3-bit quantum gates
4. all n-bit quantum gates
5. all quantum gates

[1] It is not even given in the case of a probabilistic Turing machine.

Step 1 and 2: Toffoli gate

We want to calculate powers of S_Q. The basis should be $0 = |000\rangle$, $1 = |001\rangle$, …, $6 = |110\rangle$, $7 = |111\rangle$. The $4n+1$-th power of S_Q in matrix form is

$$S_Q^{4n+1} = \begin{pmatrix} \mathbb{1}_6 & & \\ & i\cos\pi\alpha(2n+1/2) & \sin\pi\alpha(2n+1/2) \\ & \sin\pi\alpha(2n+1/2) & i\cos\pi\alpha(2n+1/2) \end{pmatrix}.$$

The $S_Q^{4n+1} = S_T$ for cos/sin-arguments $\pi(2m+1/2)$, $m \in \mathbb{N}$, in words: it is arbitrarily close to the Toffoli gate with $\pi\alpha(2n+1/2)$ for some $n \in \mathbb{N}$. The Toffoli gate is therefore in the repetoire, the proof thereof is similar to the one that powers of QM NOT are adequate to the logical NOT. Moreover, the Toffoli gate is universal for all logic gates, meaning that Q is adequate to the set of all logic gates.

Step 3: 3-bit quantum gates

Consider now powers of Q of the form $4n$ with $n \in \mathbb{N}$:

$$\begin{aligned} S_Q^{4n} &= \begin{pmatrix} \mathbb{1}_6 & & \\ & \cos 2n\pi\alpha & -i\sin 2n\pi\alpha \\ & -i\sin 2n\pi\alpha & \cos 2n\pi\alpha \end{pmatrix} \\ &\equiv \begin{pmatrix} \mathbb{1}_6 & & \\ & \cos\lambda & i\sin\lambda \\ & i\sin\lambda & \cos\lambda \end{pmatrix} \equiv U_\lambda \end{aligned}$$

These are in the repetoire, since there exists $m \in \mathbb{N}$ such that $|2\pi n\alpha - 2\pi m| < \varepsilon$ for ε arbitrarily small.

Permutations describe logic gates, and therefore in the repetoire; the limes of combinations of permutations and U does also:

$$\begin{aligned} \lim_{n\to\infty} \quad & [P_{56}(U_{\sqrt{\lambda/n}}P_{57})^2(U_{-\sqrt{\lambda/n}}P_{57})^2P_{56}]^n \\ & = \begin{pmatrix} \mathbb{1}_6 & & \\ & \cos\lambda & \sin\lambda \\ & -\sin\lambda & \cos\lambda \end{pmatrix} \equiv V_\lambda \\ \lim_{n\to\infty} \quad & [U_{\sqrt{\lambda/2n}}V_{\sqrt{\lambda/2n}}U_{-\sqrt{\lambda/2n}}V_{-\sqrt{\lambda/2n}} \\ & = \mathrm{diag}(1,\ldots,1,e^{-i\lambda},e^{i\lambda}) \equiv W_\lambda. \end{aligned}$$

A change in the global phase factor does not change the expectation value of an observable, therefore we have that

$$X_\lambda \equiv \mathrm{diag}(1,\ldots,1,e^{i\lambda}) \tag{1.11}$$

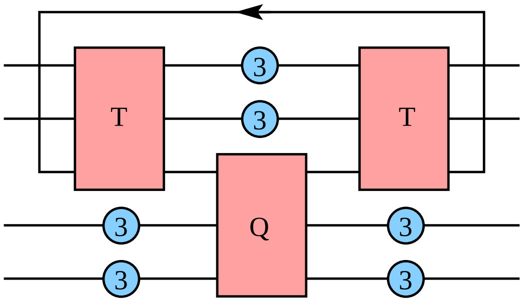

Figure 1.5: General gate with four qubits.

describes a gate that Q is adequate to. Until now, $V_\lambda, W_\lambda, X_\lambda$ are in the repetoire. Construct now a gate that maps the sixth qubit of a general input vector $|\Psi\rangle$ to zero, and puts together the two prefactors of qubit 6 and 7:

$$\begin{aligned}
|\Psi\rangle &= \sum_{n=0}^{7} c_n |n\rangle, \quad \sum_{n=0}^{7} |c_n|^2 = 1 \\
Z_6[|\Psi\rangle] &:= X_{-\arg(c_6 c_7)/2} V_{-\arctan|c_6/c_7|} W_{-\arg(c_7/c_6)/2} \\
|\Psi\rangle &\Rightarrow \sum_{n=0}^{5} c_n |n\rangle + 0 + \sqrt{|c_6|^2 + |c_7|^2} |7\rangle. \qquad (1.12)
\end{aligned}$$

The gate Z_6 is in the repetoire, since it is a combination of gates that Q is adequate to. By analogy follows that the map $G : \; c_i \rightarrow 0, i < 7$ as a gate is also in the repetoire. One can now construct a gate that evolves all coefficients from $|0\rangle, \ldots, |6\rangle$ to zero and the one from $|7\rangle$ to 1:

$$\begin{aligned}
S_{G[|\psi\rangle]} &= \sum_{n=0}^{7} e^{i\sigma_n} |\Psi_n\rangle\langle\Psi_n|; \\
S &= \prod_{n=0}^{7} S_{G^{-1}[|\Psi_n\rangle]} X_{\sigma_n} S_{G[|\Psi_n\rangle]}.
\end{aligned}$$

This last S describes the general action of a three-bit gate and is manifestly in the repetoire. So Q is universal wrt all 3×3-matrices.

Step 4 & 5: n bit gates

Look at a possible general four-bit gate as in fig. 1.5. The loopback is necessary to connect all inputs and outputs. Its input is initialized to 0. By plugging in all 2^4 different inputs it can be verified that the output of the loopback is always 0. This loopback makes the circuit reversible, the use of a source and a sink would yield irreversible gates. The action of this gate is

$$\begin{aligned}|a,b,0,c,d\rangle &\Rightarrow |a,b,ab,c,d\rangle\\ &\Rightarrow [1+abc(i\cos\pi\alpha/2-1)]|a,b,ab,c,d\rangle+[abc\sin\pi\alpha/2]|a,b,ab,c,1-d\rangle\\ &\Rightarrow [1+abc(i\cos\pi\alpha/2)]|a,b,0,c,d\rangle+[abc\sin\pi\alpha/2]|a,b,0,c,1-d\rangle.\end{aligned}$$

Its S-matrix, evaluated for the three gates, is

$$S^{abcd}_{Q_4 a'b'c'd'}=\delta^a_{a'}\delta^b_{b'}\delta^c_{c'}[(1-abc)\delta^d_{d'}+iabce^{-i\pi\alpha/2}(S^\alpha_N)^d_{d'}].$$

One can use the same procedure to get $5,6,\ldots,n$-bit gates. Therefore the n-bit gates are also in the repetoire. This closes the proof.

1.3 Quantum Turing Machine

1.3.1 Quantum Turing Machine (QTM)

The second model for quantum computing is given by the Quantum Turing Machine. As analogon to the classical TM a one-tape QTM consists of a finite processor and an infinite memory, see fig. 1.6. The head is in state $|l\rangle$ at position j of the tape and the state of the qubit at site j is denoted by $|s_j\rangle$. Computation proceeds in steps of fixed duration T. During a step only the processor and a finite part of memory interact. The QTM *halts*, if two subsequent states are identical or if the halt flag is set. The *halt flag* is an observable with spectrum $\{0,1\}$, independent of $\hat{I}$. Its state should be measurable without disturbing the state of the QTM. The QTM is universal, it can simulate any other quantum computer.

No difference to the classical TM has shown up yet. What differentiates between them is the fact that a QTM acts with the quantum state of its head on quantum states on the tape instead of logic states and can produce entanglement.

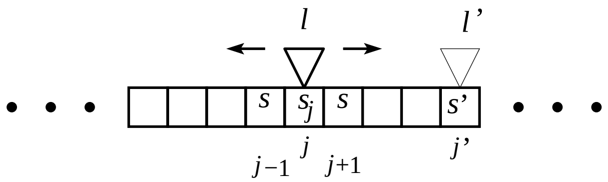

Figure 1.6: QTM

1.3.2 Church-Turing hypothesis

Does this affect some of the most fundamental statements on computing models, e.g. the Church-Turing principle? Not at all: The Church-Turing hypothesis states that every function which would naturally be regarded as computable can be computed by the universal Turing machine. Expressed as a physical principle, this reads as: Ëvery finitely realizible physical system can be perfectly simulated by a universal computing machine operating by finite means.¨
The QTM fulfills this principle, and the original hypothesis as well. A classical Turing machine does *not* fulfill the second version, since it is finite, but continuous systems can be described with only a few parameters.

1.3.3 Step Operator

Now that we know what models do exist, how can we describe the action of them? How do they evolve with time? The most basic step of a calculation is described by a *step operator*. The head of a QTM interacts with the tape only at one position in a fixed time. It can move to the left, to the right, or stay and interact. All this can be described with a unitary *step operator* T. It must be local and may describe a displacement in at most one direction. Moreover, the periodicity of the lattice sites must be taken into account. Mathematically these three requirements are expressed as

$$\begin{aligned} \langle l', j', s'|T|l, j, s\rangle &= \langle s'_{\neq j}|s_{\neq j}\rangle\langle l', j', s'_{j'}|\tilde{T}|l, j, s_j\rangle, \\ \tilde{T} &= \sum_{j=-\infty}^{\infty}\sum_{\Delta=-1}^{1} P_{j+\Delta}\tilde{T}P_j, \\ \langle l', j'+\Delta, s'|\tilde{T}|l, j', s\rangle &= \langle l', j+\Delta, s'|\tilde{T}|l, j, s\rangle. \end{aligned}$$

1.3.4 Dynamics

The dynamics of a QTM can be described by its Hamiltonian. For a gate, this is according Deutsch,

$$H \equiv \frac{i}{t}\ln S. \tag{1.13}$$

H defined in such a way is local; the description complexity is relatively small. Feynman proposed another form of H for general QTMs with step operator T,

$$H \equiv K(2 - T - T^{\dagger}). \tag{1.14}$$

This gives the kinetic energy, if T is a simple displacement, without interaction. T by itself can be a sum of elementary unitary step operators for single gates and therefore describes the evolution of a whole circuit.

1.4 Complexity

One can estimate the cost of a computation for a QTM, i.e. the resources time, memory space and energy in the framework of complexity theory. What measures do exist?
The *size* of a circuit gives the number of elementary gates in a quantum circuit. The *depth* is the maximal length of a directed path from in- to output register. An *interacting pair of quantum circuits* describes a partition of the circuit with disjoint sets of inputs such that all outputs are on one side. The *communication cost* gives then the number of wires between interacting pairs. A quantum circuit (n, t)-*simulates* a QTM, if input $\tilde{x} \in \{0, 1\}^n$ evolved by C is the same as the state of M after t steps.

Theorems by Yao

1. Any unitary operator $U \in \mathbb{C}^{2^n}$ can be simulated by a quantum network using $2^{\mathcal{O}(n)}$ 3-gates, with $\mathcal{O}(n)$ wires.

2. Every QTM can be (n, t)-simulated by a quantum network of size $poly(n, t)$.

3. There exists a universal QTM that can simulate any other QTM with only polynomial slowdown

See Yao 1993 [4] for a proof of these theorems.

1.4.1 QTM vs. TM: computation speed

A QTM is not faster than a classical Turing machine on average if it is performing simple calculations. One could imagine an algorithms taking advantage of ”parallel universes” to instantiate copies of the QTM and return the result in a shorter time $t = pt_0, p < 1$, but with a probability of p only to return a result. There is a huge speed-up, however, if specialized algorithms like the ones of Deutsch-Josza and Grover are considered. See later contributions for more details.

1.5 Summary

Quantum gates have qubits as inputs, that is superpositions of states. The Q-gate is universal wrt the set of all quantum gates. The proof thereof constructs a repetoire of gates that Q is adequate to. The QTM is constructed as a quantum analogon to the Turing machine. It fulfills the Church-Turing principle and can be described by step operators or a Hamiltonian. The simulation of a QTM is possible by a quantum circuit or another QTM with polynomial slowdown. A quantum computer c̈alculates in parallel universes; not faster in average.

Bibliography

[1] D. Deutsch, *Quantum theory, the Church-Turing principle and the universal quantum computer*, Proc. R. Soc. Lond. A **400**, 97 (1985).

[2] D. Deutsch, *Quantum computation networks*, Proc. R. Soc. Lond. A **425**, 73 (1989).

[3] P. Benioff, *Models of quantum turing machines*, Fortsch. Phys. **46**, 423 (2007).

[4] A. C. Yao, *Quantum Circuit Complexity*, Proc. of the 34th Ann. Symp. on Found. of Comp. Sc. (FOCS) p. 352 (1993).