Quantum Computing

Proseminar in Theoretical Physics
Institut für theoretische Physik
ETH Zürich

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

FS08

typeset, compilation & editing by H. G. Katzgraber

Table of Contents

TABLE OF CONTENTS

Chapter 1

Computational Models for Quantum Computing

Pascal Stephan Philipp Steger
supervisor: Dr. Stefan Hohenegger

We will give an overview over the quantum analogues to the classical computing models, the Turing machine and the circuit model. The two basic models for quantum computing are developed: The quantum computing network or circuit, built of quantum gates, and the Quantum Turing Machine (QTM). The differences between the classical and quantum concepts are highlighted.
Connections between the two models are established by the fact that a gate is conveniently described by a QTM and every QTM can be simulated by a quantum network; we show a close relation between the resources needed by the QTM and the network simulation. The description of the dynamics using step operators T or S-matrices treats the QTM and gates in a similar way. Advantages over the classical computing models are mentioned.

1.1 Introduction

Previous contributions have introduced the basic notions for classical computing: The Turing machine as well as the circuit model emerged as classical computing models, complexity classes and their use in cryptography were outlined. We want to look at the generalization of these models to the framework of quantum mechanics. One motivation for this is that the security of several cryptographic algorithms relies on long calculation times. The latter may be shortened using

quantum computers. The search for prime factors of a big number, for instance, could be replaced with more efficient quantum versions.
We start with the circuit model in section 2.1, introducing quantum gates with examples and giving an efficient mathematical description. As we will see, one cannot perfectly imitate a classical gate using quantum gates, but an arbitrarily close approximation is possible. A constructive proof thereof follows in 2.4. The QTM will be dealt with in section 3, followed by its description with a step operator. The two models are related to each other, as we shall see in section 4, and their dynamics may be described easily with a Hamiltonian. A last part is dedicated to the possible gain in computation speed with quantum computers.

1.2 Quantum Gates

1.2.1 Terminology

The well-known classical computer performs computations using logic circuits. Let us study this in some detail, introducing some later needed terminology: A *computation* is a process that produces output depending on some input. *Input* and *output* denote abstract symbols. They are encoded in *bits* or *quanta*, which are the smallest possible quantities of non-probabilistic information. Those again are physically represented in a carrier, e.g. a transistor, or a spin 1/2-particle.
Physical processes in a quantum computer follow three steps:

1. preparation of the input states in carriers, e.g. setting the spin of electrons
2. interaction in QM elastic scattering
3. measurement of output carriers after a fixed number of steps

For most applications, one can neglect the details of step 2. It 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 the theoretical model. The models should take into account, although, that errors may occur. Error correction will be the main topic of a following contribution.
What is a *logic gate*? It is a computing machine, where input and output consist of fixed number of bits. Some previously defined computation is done in a fixed time. A *quantum gate* on the other hand can have qubits as input and output. These are quantum mixtures of eigenstates of the input observable $\hat{I}$ and output observable $\hat{O}$. 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 Landauer's principle [1] does not give an energy loss.

Reversible gates and circuits must have the same number of input and output wires. Any irreversible gate can be converted to a reversible one by repeating an appropriate number of inputs on the output side.

One can connect several gates into a circuit. The outputs of a gate after com-

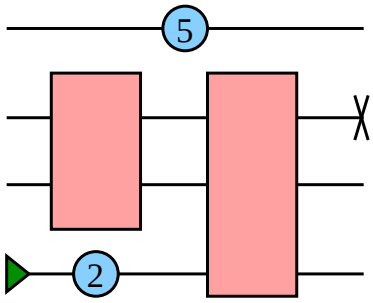

Figure 1.1: Example for a logic circuit showing different pieces.

putation step (i) can be used as inputs for another gate at step $(i+1)$, requiring that the gates are synchronized. A *logic circuit* is a computing machine consisting of logic gates, a computation is performed in fixed time. The symbols in the example circuit – see fig. 1.1 – are place holders for different parts of a general circuit: Gates are denoted by rectangles with input connections to the left and output connections to the right. Information flows from the left to the right. Sources and sinks, graphically represented by triangles and crosses, can be used to implement special input and output conditions: A *source* is has only one output that emits 0 or 1 in each step. A sink on the other hand has only one input and irreversibly deletes information. A *unit wire* propagates the carriers unchanged and computes the identity function, a fixed time dilation is indicated by a number in a circle.

1.2.2 Mathematical Descriptions

Several mathematical descriptions of gates are possible. We first have to choose a *computational basis*, which is given by the eigenstates of the input operator $\hat{I}$ and output operator $\hat{O}$ in the Schrödinger picture. They should be the same, otherwise it would be non-trivial to use an output of a gate as input for the next one; repeated gates could not be implemented easily.

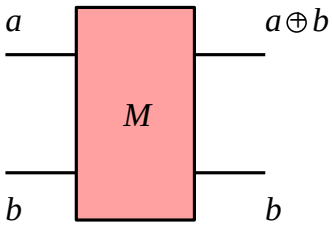

Figure 1.2: measurement gate - XOR - CNOT

Tables

In this basis one can write down the action of a gate using a table. Every combination of input eigenvalues is mapped to its output. 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)

The classical gate is called XOR since it computes the logical XOR function from inputs a and b, copying a as a second output to guarantee reversibility. The action of the quantum gate can be seen as inverting the b input if a is set and returning b unchanged otherwise. Therefore it is referred to as CNOT – controlled NOT – or measurement gate.

Permutations

Another representation of the same description is possible using permutations: let $\{|a, b\rangle\},\ a, b \in \{0, 1\}$ be the four computational basis states for a system with two inputs and outputs. A gate maps each input state to an output state, for the measurement gate this reads as

$$\begin{aligned} |0,0\rangle &\rightarrow |0,0\rangle \\ |0,1\rangle &\rightarrow |1,1\rangle \\ |1,0\rangle &\rightarrow |1,0\rangle \\ |1,1\rangle &\rightarrow |0,1\rangle. \end{aligned} \tag{1.2}$$

S-matrix

A third way is given by using an S-Matrix, which is most suitable for quantum gates, as was shown by Deutsch [2]. In principle, it encodes the mapping from

above with a unitary matrix, thus using linear algebra.
Let us consider the measurement gate. Its S-matrix is given by

$$S^{ab}_{a'b'} = \delta^{a\oplus b}_{a'}\delta^{b}_{b'}: \qquad \begin{pmatrix}|0,0\rangle\\|0,1\rangle\\|1,0\rangle\\|1,1\rangle\end{pmatrix} = \begin{pmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{pmatrix}\begin{pmatrix}|0,0\rangle\\|0,1\rangle\\|1,0\rangle\\|1,1\rangle\end{pmatrix}. \tag{1.3}$$

This unitary matrix $S^{ab}_{a'b'}$ has clumped indices ab, $a'b'$ denoting eigenstates of the input and output carriers. $a, b \in \{|0\rangle, |1\rangle\}$ results in four possible combinations, the same holds for the two outputs a', b'. We can interprete the clumped indices as binary representations of a natural number, as for instance $ab = 10\text{b} = 2$. In this interpretation, they give the position of the corresponding matrix element in S. As an example, we have for the element in the first row and the second column

$$a = 0,\, b = 0,\, a' = 0,\, b' = 1: \quad S^{a'b'}_{ab} = \delta^{0\oplus 0}_{0}\delta^{0}_{1} = 0. \tag{1.4}$$

The operation of a general gate with n in- and outputs corresponds to a matrix multiplication with $S^{ab\ldots}_{a'b'\ldots}$,

$$|a, b, \ldots\rangle \to \sum_{a',b',\ldots\in\{0,1\}} S^{ab\ldots}_{a'b'\ldots}|a', b', \ldots\rangle \equiv S|a, b, \ldots\rangle. \tag{1.5}$$

If no basis is chosen explicitly, S denotes the linear operator of a gate. Unitarity of S is necessary to describe a reversible gate. Repeated gates are represented by powers of S, for instance, the unit wire with time dilation n can be interpreted as n individual unit wires with a constant time dilation. Its action is $\mathbb{1}^n = \mathbb{1}$.

1.2.3 EXAMPLES OF QUANTUM GATES

EXAMPLE: NOT

For an more complicated example of the action of the S-matrix, consider the NOT gate in fig. 1.3. It is described by

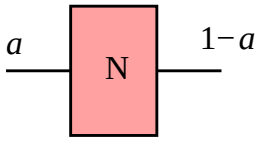

Figure 1.3: NOT gate

$$\begin{array}{c|c} a & \neg a \\ \hline 0 & 1 \\ 1 & 0 \end{array}, \qquad \begin{aligned} |0\rangle &\to |1\rangle \\ |1\rangle &\to |0\rangle, \end{aligned} \qquad S_N = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{1.6}$$

as one can check by multiplication with $(0,1)^T$ or $(1,0)^T$, which are vectors denoting the states $|1\rangle$ and $|0\rangle$ in the computational basis $\{|0\rangle, |1\rangle\}$. Powers of S correspond to several successive copies of NOT, $\alpha \in \mathbb{N}$ implies that N^α is a logic gate, either the identity (α even) or the NOT gate (α odd). Nevertheless, a non-integer power $\alpha \notin \mathbb{N}$ of the operator N is perfectly well defined as well: N^α does then not describe the action of a logic gate anymore, but the action of a *quantum gate*. We can compute the S-matrix

$$S_{N^\alpha} = S_N^\alpha = \frac{1}{2}\begin{pmatrix} 1+e^{i\pi\alpha} & 1-e^{i\pi\alpha} \\ 1-e^{i\pi\alpha} & 1+e^{i\pi\alpha} \end{pmatrix}. \tag{1.7}$$

Here we see that the use of states as in-/outputs instead of simple logic 0 or 1 is justified. A gate then transforms these states into another, usually visualized as rotation on the Bloch sphere [3].

Toffoli and Q

The Toffoli gate from classical computing (see fig. 1.4) has an analogue in quantum computing. Its classical version can be described by the S-matrix

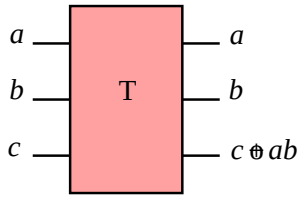

Figure 1.4: Toffoli gate

$$S^{abc}_{Ta'b'c'} = \delta^a_{a'}\delta^b_{b'}[(1-ab)\delta^c_{c'} + ab(S_N)^c_{c'}]. \tag{1.8}$$

Analogously, the quantum gate Q reads as

$$S^{abc}_{Qa'b'c'} = \delta^a_{a'}\delta^b_{b'}\left[(1-ab)\delta^c_{c'} + iabe^{-i\pi\alpha/2}(S^\alpha_N)^c_{c'}\right]. \tag{1.9}$$

and boils down to the classical form if $\alpha \in \mathbb{N}$.
In order to calculate individual matrix elements, we choose the basis $0 = |000\rangle$,

$1 = |001\rangle$, ..., $6 = |110\rangle$, $7 = |111\rangle$. The S-matrices are then computed by plugging in the different values for $abc, a'b'c'$,

$$S_T = \begin{pmatrix} \mathbb{1}_6 & & \\ & 0 & 1 \\ & 1 & 0 \end{pmatrix}, \tag{1.10}$$

$$S_Q = \begin{pmatrix} \mathbb{1}_6 & & \\ & i\cos\pi\alpha/2 & \sin\pi\alpha/2 \\ & \sin\pi\alpha/2 & i\cos\pi\alpha/2 \end{pmatrix}. \tag{1.11}$$

$\mathbb{1}_6$ denotes a 6×6-identity matrix. The classical S_T-matrix can be checked by looking at the tabular description.

1.2.4 Equivalence

A question that may arise is whether it is possible to simulate a classical logic gate by using a set of quantum gates: Consider for example the repeated use of a NOT-gate,

$$\begin{aligned} S_{N^2} &= S_N^2 = \mathbb{1}, \\ (S_{N^\alpha})^m &= S_N^{m\alpha} = S_N^{m\alpha - 2\lfloor m\alpha/2 \rfloor}. \end{aligned} \tag{1.12}$$

The exponent $m\alpha - 1\lfloor m\alpha/2 \rfloor \equiv 1+\varepsilon$ can be made arbitrarily close to 1, but never exactly for $m \in \mathbb{N}$ and an irrational α. A failure of the simulation is possible, namely if another result than the classically expected one shows up. The time before non-classical behavior is given by the reciprocal of the expectation value for the wrong result,

$$t = \frac{1}{\max_{|\Psi\rangle}(1 - |\langle\Psi|S_N^\dagger S_{N^\alpha}^m|\Psi\rangle|^2)} = \frac{1}{\sin^2 \pi\varepsilon/2} \sim \varepsilon^{-2} \overset{(\varepsilon\to 0)}{\to} \infty. \tag{1.13}$$

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

$$\lim_{n\to\infty} S_{g_n} e^{i\phi_n} = S_f. \tag{1.14}$$

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

[1] It is not even given in the case of a probabilistic Turing machine. A redefinition of equivalence with fixed probabilities would solve this problem.

of unit wire, source and this gate is adequate to the set of all possible gates. The Toffoli gate is universal for classical gates. We will see that the Q-gate plays the same role for quantum gates:

Claim:

The Q-gate is a universal gate.

Proof:

A constructive proof is striven for; we create a repertoire 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

This procedure was proposed by Deutsch [2].

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

$$S_Q^{4n+1} = \begin{pmatrix} \mathbb{1}_6 & & \\ & i\cos\pi\alpha(2n+1/2) & \sin\pi\alpha(2n+1/2) \\ & \sin\pi\alpha(2n+1/2) & i\cos\pi\alpha(2n+1/2) \end{pmatrix}. \tag{1.15}$$

S_Q^{4n+1} equals S_T for $\alpha(2n+1/2) = (2m+1/2)$, for some $m \in \mathbb{N}$. For the same reason that powers of QM NOT are adequate to the logical NOT – there exists an arbitrarily close approximation – the Toffoli gate is in the repertoire. Moreover, the Toffoli gate is universal for all logic gates, so Q is also adequate to the set of all logic gates.

STEP 3: 3-BIT QUANTUM GATES

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

$$\begin{aligned} S_Q^{4n} &= \begin{pmatrix} \mathbb{1}_6 & & \\ & \cos 2n\pi\alpha & -i\sin 2n\pi\alpha \\ & -i\sin 2n\pi\alpha & \cos 2n\pi\alpha \end{pmatrix} = \\ U_\lambda &\equiv \begin{pmatrix} \mathbb{1}_6 & & \\ & \cos\lambda & i\sin\lambda \\ & i\sin\lambda & \cos\lambda \end{pmatrix}. \end{aligned} \tag{1.16}$$

These are in the repertoire, 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 (e.g. P_{57}, it permutes qubits 5 and 7), and belong therefore to the repertoire; the limit of combinations of permutations and U_λ as well:

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

A change in the global phase factor does not change the expectation value of an observable, so

$$X_\lambda \equiv \mathrm{diag}(1,\ldots,1,e^{i\lambda}) \tag{1.17}$$

describes a gate that Q is adequate to. Until now, we know that $V_\lambda, W_\lambda, X_\lambda$ are all in the repertoire. We can then construct 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:

$$\begin{aligned} |\Psi\rangle &= \sum_{n=0}^{7} c_n|n\rangle, \quad \sum_{n=0}^{7}|c_n|^2 = 1 \\ Z_6[|\Psi\rangle] &:= X_{-\arg(c_6c_7)/2}V_{-\arctan|c_6/c_7|}W_{-\arg(c_7/c_6)/2} \\ |\Psi\rangle &\Rightarrow \sum_{n=0}^{5} c_n|n\rangle + 0 + \sqrt{|c_6|^2+|c_7|^2}|7\rangle. \end{aligned} \tag{1.18}$$

The gate Z_6 is in the repertoire, since it is a combination of gates that Q is adequate to. It follows by analogy that the same map for another qubit $i < 7$, namely $G: c_i \to 0$ – and the gate it is describing – is also in the repertoire. One

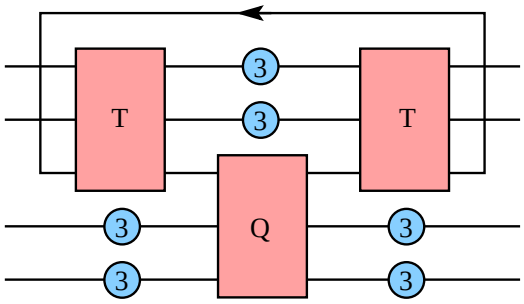

Figure 1.5: General gate with four qubits.

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:

$$\begin{aligned} S_{G[|\psi\rangle]} &= \sum_{n=0}^{7} e^{i\sigma_n} |\Psi_n\rangle\langle\Psi_n|; \\ S &= \prod_{n=0}^{7} S_{G^{-1}[|\Psi_n\rangle]} X_{\sigma_n} S_{G[|\Psi_n\rangle]}. \end{aligned} \tag{1.19}$$

This last S describes the general action of a three-bit gate and is manifestly in the repertoire. So Q is universal with relation to the set of all 3×3-matrices.

Step 4 & 5: n bit gates

Look at a possible general four-bit gate as in fig. 1.5. The loop-back is necessary to connect all inputs and outputs[2]. Its input is initialized to 0. By plugging in all 2^4 different inputs it can be verified that the output of the loop-back is always 0. The action of this gate yields

$$\begin{aligned} |a,b,0,c,d\rangle &\Rightarrow |a,b,ab,c,d\rangle \\ &\Rightarrow [1+abc(i\cos\pi\alpha/2-1)]|a,b,ab,c,d\rangle \\ &\quad +[abc\sin\pi\alpha/2]|a,b,ab,c,1-d\rangle \\ &\Rightarrow [1+abc(i\cos\pi\alpha/2)]|a,b,0,c,d\rangle + [abc\sin\pi\alpha/2]|a,b,0,c,1-d\rangle. \end{aligned} \tag{1.20}$$

Its S-matrix, evaluated for the three gates, reads as

$$\mathcal{S}^{abcd}_{Q_4 a' b' c' d'} = \delta^a_{a'}\delta^b_{b'}\delta^c_{c'}\left[(1-abc)\delta^d_{d'} + iabce^{-i\pi\alpha/2}(S^\alpha_N)^d_{d'}\right]. \tag{1.21}$$

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

[2]This loop-back makes the circuit reversible, the use of a source and a sink would yield irreversible gates.

1.3 Quantum Turing Machine (QTM)

1.3.1 Connection to the Classical Turing Machine

The second model for quantum computing is given by the Quantum Turing Machine, described in detail by Benioff [4]. In direct analogy to the classical one-tape Turing Machine, a one-tape QTM consists of

- an infinite memory,
- a finite processor,
- a state control and
- a program.

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 $|s_j\rangle$, see fig. 1.6. The computation proceeds in steps of fixed duration Δt. 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 $\{0, 1\}$, its state should be measurable without disturbing the state of the QTM. The QTM is universal, it can simulate any other quantum computer.

The main difference between the classical TM and the QTM is the fact that a

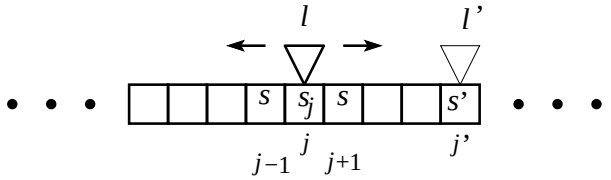

Figure 1.6: Sketch of a one-tape QTM. Program and state control are part of the head.

QTM acts with the quantum state of its head on quantum states on the tape instead of logic states. Any superposition is allowed at a given lattice site, therefore much more information than only the boolean 0 or 1 can be stored.

1.3.2 Church-Turing hypothesis

Does the existence of quantum superpositions on the tape 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:

> *Every finitely realizable 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 may be described with a finite number of parameters only.

1.3.3 Step Operator

Since the QTM operates in steps of finite time, it is a good idea to define a unitary *step operator* T, which must fulfill the following requirements: It should describe the interaction of the head with the tape only at one position per time interval. The head can move to the left, to the right, or stay and interact. 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 the last three requirements are expressed in eq. 1.22. The operator $\tilde{T}$ describes that part of the step operator T involved in the interaction of the head with a single lattice qubit. $P_j = |j\rangle\langle j|$ is a projection operator for the head onto the lattice site j. Δ stands for mere displacements – to the left, not at all, to the right. It can take the respective values $-1, 0, 1$.

$$\begin{aligned} \langle l', j', s'|T|l, j, s\rangle &= \langle s'_{\neq j}|s_{\neq j}\rangle\langle l', j', s'_{j'}|\tilde{T}|l, j, s_j\rangle, \\ \tilde{T} &= \sum_{j=-\infty}^{\infty}\sum_{\Delta=-1}^{1} P_{j+\Delta}\tilde{T}P_j, \\ \langle l', j'+\Delta, s'|\tilde{T}|l, j', s\rangle &= \langle l', j+\Delta, s'|\tilde{T}|l, j, s\rangle. \end{aligned} \tag{1.22}$$

The first equation states that a change in a single step can only interact with the qubit at the position of the head. All other contributions are excluded by $\langle s'_{\neq j}|s_{\neq j}\rangle = 0$. The second equation expresses the motion in only one direction, $\tilde{T} \neq 0$ only if the motion takes the head from position j to position $j + \Delta$, such that $P_{j+\Delta}$ is not orthogonal to $\tilde{T}P_j$. The last equation shows that the matrix elements of interaction are the same for all different j and j', which is only possible if the lattice sites are periodic.

1.4 Connections

The computing models QTM and quantum circuits can be transformed into each other. One can get quantitative statements for the efficiency of the respective simulation from complexity theory. This theory is needed to determine the cost of a computation in general, i.e. the resources time, memory space and energy. 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 located on one side. The *communication cost* gives then the number of wires between interacting pairs. See fig. 1.7 for an example. These measures can be used to characterize a quantum circuit, especially when optimization is considered. If one wants a fast computer, the depth and the communication cost should be small – information propagates with the speed of light at most, so fewer and shorter wires mean faster computation. If the circuit is optimized for small space, its size must be minimized. The

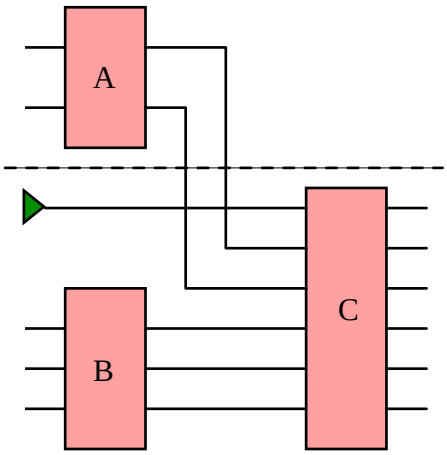

Figure 1.7: Example for a reversible interacting pair of quantum circuits. Its size is 3, the depth is also 3, the communication cost equals 2.

simulation of a given QTM by a quantum circuit could be polynomially bounded or increase exponentially. In order to distinguish between these two cases, we introduce the (n,t)-simulation: A quantum circuit C (n,t)-*simulates* the QTM M, if the input $\tilde{x} \in \{0,1\}^n$ evolved by C is the same as the state of M after t steps, provided that the first n qubits of the QTM are assigned the same input values as the circuit.

1.4.1 Theorems by Yao

Yao [5] gives, between others, following theorems that relate the QTM and implementations on a quantum circuit:

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.

The proofs are shown in the article by Yao, and go beyond the scope of this introduction.

1.4.2 Dynamics

The dynamics of a QTM or a quantum circuit can be described by their Hamiltonian. For a single gate with a given S-matrix, the Hamiltonian can be constructed in the following way, as Deutsch [2] outlined:

$$H \equiv \frac{i}{t} \ln S, \tag{1.23}$$

where the logarithm of the gate's S-matrix evaluated by a Taylor series. The Hamiltonian H for a quantum circuit can then be constructed by connecting single-gate Hamiltonians.
Feynman [6] proposed to use the step operator T to get the Hamiltonian for general QTMs,

$$H \equiv K(2 - T - T^\dagger). \tag{1.24}$$

This expression gives the kinetic energy, if T is a simple displacement without interaction. Here another connection between the two basic models shows up: T by itself can be a sum of elementary unitary step operators for QTMs describing single gates, and therefore T incorporates all information of the evolution of all the circuit, as if it were a QTM.

1.4.3 QTM vs. TM: computation speed

A QTM is not faster than a classical Turing machine on average if it is using the same algorithms. Deutsch [7] for instance considered a computer that needs one day to predict the stock market of tomorrow. If its program was implemented

within a quantum computer, it would be possible to stop the computation after half a day and get the result already, but there is a probability of 50% that no result shows up at all. One could imagine that this quantum computer takes 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 failure probability of $1 - p$. However, there is an average huge speed-up, if specialized algorithms like the ones of Deutsch-Josza [8] and Grover [9] are considered. These do not destroy the superpositions of states during calculation and project only at the end of the computation. See later contributions for more details.

1.5 Summary

In this outline we have considered three models for quantum computing:
Quantum gates take qubits as input, let them interact in a QM elastic scattering and measure them at the end. The direct way from in- to output is best described using an S-matrix, which is unitary for reversible gates. The repeated application of NOT-gates motivates the use of quantum gates. Exact equivalence of quantum gates with relation to logic gates is not possible, but approximation is. The Q-gate is universal to the set of all quantum gates, as the Toffoli is for all logic gates. Different combinations of powers of Q, permutations and phase factor changes are used for the proof.
The QTM uses quantum states on lattice sites and in the head, but the rest of its components are equivalent to the classical Turing machine. It fulfills the Church-Turing principle. The description with a step operator reflects that the QTM performs computations in several steps. Complexity theory characterizes quantum computers by their use of space, memory, and time. QTM and quantum circuit can simulate each other with only polynomially increased need in memory and time. Hamiltonians for the descriptions of the dynamics make use of the S-matrix and step operator T. A quantum computer can reduce the time needed for special algorithms, but for classical algorithms it takes the same time on average.

Acknowledgments

I owe Dr. St. Hohenegger a big "thank you" for his helpful advice on how to present this most interesting topic in a stringent and reader-friendly way. I would also like to thank Prof. Dr. H. Katzgraber and Prof. Dr. R. Renner for creating a pleasant atmosphere at the proseminar.

Bibliography

[1] R. Landauer, *Uncertainty Principle and Minimal Energy Dissipation in the Computer*, Int. J. Theor. Phys. **21**, 283 (1982).

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

[3] I. Chuang, *Quantum physics and Church-Turing* (http://http.cs.berkeley.edu/∼vazirani/f97qcom/, 1997).

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

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

[6] R. P. Feynman, *Quantum mechanical computers*, Opt. News **11**, 11 (1985).

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

[8] D. Deutsch and R. Josza, *Rapid solutions of problems by quantum computation*, Proc. R. Soc. London A **439**, 553 (1992).

[9] L. Grover, *Quantum mechanics helps in searching for a needle in a haystack*, Phys. Rev. Lett. **79**, 325 (1997).