Evolving Blackbox Quantum Algorithms using Genetic Programming

Dissertation

submitted to the
Department of Physics
University of Dortmund
Germany

for the degree of
Doctor of Natural Sciences
(Dr. rer. nat.)

presented by
Dipl.-Phys. Ralf Stadelhofer

Dortmund, May 2006

Contents

Symbols and Abbreviations

Symbol	Explanation
$\mathbb{R}$	Set of real numbers.
$\mathbb{N}$	Set of integer numbers.
$\mathbb{C}$	Set of complex numbers.
$\partial_x f(x)$	Partial derivative of the scalar function $f(x)$ with respect to the scalar x.
$\underline{\partial_x} f(\underline{x})$	Gradient of the multivariable scalar function $f(\underline{x})$ with respect to the vector $\underline{x}$.
$(\underline{\partial_x})^2 f(\underline{x})$	Laplace operator on the multivariable scalar function $f(x)$ with respect to x.
$\psi(\underline{x})$	Scalar wave function on the configuration space $\underline{x}$.
$\|\psi\rangle$	Quantum state.
$\|\psi]$	Classical state that does not allow for superpositions.
$\boldsymbol{\varrho}$	Density matrix.
$\boldsymbol{\varrho}_{therm}$	Thermal density matrix.
$\underline{\varphi}(\underline{x})$	Vector-valued wavefunction on the configuration space $\underline{x}$.
$\hbar$	$h/(2\pi)$.
$\mathbf{1}$	Identity operator.
$\langle a, b\rangle$	Scalar product of the vectors a and b.
$\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z$	Unit vectors along the x, y, z-axis.
$(x, y, z)^t$	Transposed of vector (x, y, z): $(x, y, z)^t = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$.
$\underline{\boldsymbol{\sigma}} = (\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y, \boldsymbol{\sigma}_z)$	Pauli matrices.
$\|\boldsymbol{U}\|$	Norm of the unitary transformation $\boldsymbol{U}$.
$\langle \boldsymbol{M} \rangle$	Expectation value of the measurement operator $\boldsymbol{M}$.
$\langle \boldsymbol{M} \rangle_{\boldsymbol{U}}$	Expectation value of the measurement operator $\boldsymbol{M}$ when the initial state was transformed according to the unitary transformation $\boldsymbol{U}$.
$\boldsymbol{\mathcal{E}}(\boldsymbol{\varrho})$	Quantum operation on the quantum state $\boldsymbol{\varrho}$.

ϵ	Error
$(\boldsymbol{\varrho})_{ij}$	Entry in the ith row and the jth column of the matrix $\boldsymbol{\varrho}$.
γ	Gyromagnetic ratio.
ω_{rf}	Frequency of a radiofrequency field.
$\omega_0^{(i)}$, $\omega_1^{(i)}$	Lamor frequencies of the ith nucleus.
ω_{ij}, $\nu(i,j)$	Frequency emitted in a transition from the ith to the jth energy state.
$(\pi)_y^H - (\pi)_x^H$	NMR pulse sequence: Apply a π pulse along the y-axis and then along the x-axis on the hydrogen nucleus.
$[n_{00}, \ldots, n_{jj}]$	Matrix that only has diagonal elements, namely n_{00} in the upper left, n_{jj} in the lower right.
$\Im(z)$	Imaginary part of the complex number z.
$\mathcal{I}$	Index set.
$\lvert\underline{x}\rvert$	Number of bits in the binary string $\underline{x}$.
$\underline{i} \oplus K$	Coset of the set K.
$\lvert\mathcal{I}\rvert$	Number of elements of the set $\mathcal{I}$.
Σ	Alphabet, e.g. binary alphabet: $\Sigma = \{0,1\}$.
Σ^n	Set of all words of length n over the alphabet Σ.
Σ^*	Set of all words over the alphabet Σ.
$\oplus$	Logical XOR operation (addition modulo two).
$\vee$	Logical AND operation.
$\wedge$	Logical OR operation.
$\neg$	Logical NOT operation.
$\mathcal{O}(g)$	Upper bound is proportional to function g.
$\Omega(g)$	Lower bound is proportional to function g.
$\Theta(g)$	Asymptotic bound is the function g.
$\lvert a\rvert$	Absolute value of variable a.
$tr\{\boldsymbol{A}\}$	Trace of the operator $\boldsymbol{A}$.
$\boldsymbol{A} \otimes \boldsymbol{B}$	Tensor product of the operators $\boldsymbol{A}$ and $\boldsymbol{B}$.
$\boldsymbol{A}^{\otimes n}$	$\underbrace{\boldsymbol{A} \otimes \cdots \otimes \boldsymbol{A}}_{n \text{ times}}$
$\underline{B}$	Magnetic field vector.
BPP	Bounded-error probability in polynomial time.
BQP	Quantum pendant to the complexity class BPP.
BUPQC	Bounded-error uniform polynomial quantum circuit.
c	Vacuum speed of light.
c_n	Number of elementary gates of the circuit C_n.
C_n	Uniform Boolean circuit on n input bits.
$\boldsymbol{CNOT}$	CNOT operator.
$d(\boldsymbol{A}, \boldsymbol{B})$	Distance between operator $\boldsymbol{A}$ and $\boldsymbol{B}$.
DDT	Deterministic decision tree.
$\deg(f)$	Degree of the polynomial that represents the Boolean function f.
$\widetilde{\deg}(f)$	Minimum degree among all polynomials that approximate the Boolean function f.
$\mathrm{D}(f)$	Deterministic decision tree complexity.
DTM	Deterministic Turing machine.
E	Energy.
$E(n), E_n$	Energy of an quantum system in its nth stationary state.

EPR	Einstein-Podolsky-Rosen.
EQP	Quantum pendant to the complexity class P.
EUPQC	Exact uniform polynomial quantum circuit.
FID	Free induction decay.
GP	Genetic Programming.
h	Planck's constant.
H	Hamilton function.
$\boldsymbol{H}$	Hamilton operator or Hadamard operation.
$\mathcal{H}$	Hilbert space.
$\boldsymbol{\mathcal{H}}$	Hamilton operator.
$H.c.$	Hermitic conjugated.
$\imath$	$\sqrt{-1}$.
$\underline{i}$	Binary decomposition of the integer i.
$\underline{i} \cdot \underline{j}$	Inner product of the binary strings $\underline{i}$ and $\underline{j}$ modulo 2.
$\boldsymbol{I}_{\alpha}^{(A)}$	αth component of the spin operator of nucleus A.
J	Scalar coupling constant.
k	Boltzmann's constant.
K	Subset of $\mathcal{I}$: $K \subset \mathcal{I}$.
L	Lagrangian, or Language: A language L is a subset of Σ^*
$\boldsymbol{L}$	Lagrangian operator.
m	Mass or an integer.
$\boldsymbol{M}$	Measurement operator.
N	Number of states of an n-qubit system: $N = 2^n$.
n	Number of qubits if not stated otherwise.
$\underline{n}$	Unit vector.
NMR	Nuclear magnetic resonance.
NMR-QC	Liquid state NMR quantum computer.
NP	Nondeterministic polynomial time complexity.
$\boldsymbol{O}$	Oracle operator.
P	Polynomial time complexity.
p, *prob*	Probability.
$P(\underline{n}_1, \underline{n}_2)$	Correlation function.
PDT	Probabilistic decision tree.
$poly(x)$	Polynomial over variable x.
POVM	Positive operator-valued measure.
PSPACE	Polynomial space complexity.
PTM	Probabilistic Turing machine.
$\mathrm{Q}_2(f)$	Double bounded error quantum decision tree complexity.
$\mathrm{Q}_E(f)$	Exact quantum decision tree complexity.
QA	Quantum algorithm.
QC	Quantum circuit.
QDT	Quantum decision tree.
QTM	Quantum Turing machine.
$\mathrm{R}_2(f)$	Probabilistic decision tree complexity.
$\boldsymbol{R}_x(\alpha)$, $\boldsymbol{R}_y(\alpha)$, $\boldsymbol{R}_z(\alpha)$	One-qubit rotations.
$\boldsymbol{R}_{zz}(\alpha)$	Two-qubit rotation.
rf	Radiofrequency
s_{FID}	Signal of the free induction decay.
$S(\Omega)$	Fourier transformed of s_{FID}: $S(\Omega) = \int_{\Omega} s_{\mathrm{FID}} e^{\imath\Omega}\, d\Omega$.

t	Time.
T	Temperature.
TM	Turing machine.
u_i	Square integrable function.
$\boldsymbol{U}$	Unitary transformation.
$\boldsymbol{U}_{\text{CNOT}}^{(ij)}$	***CNOT*** operation with i the control and j the target qubit.
$\boldsymbol{U}^t$	Transposed unitary operation $\boldsymbol{U}$: $(\boldsymbol{U})_{ij}^t = (\boldsymbol{U})_{ji}$.
$\boldsymbol{U}^+$	Adjoint of the unitary transformation $\boldsymbol{U}^+ = (\boldsymbol{U}^t)^*$.
$U(x)$	Potential energy.
X	Blackbox.
$\underline{x}$	Vector or binary string.
$\overline{x}$	Negation of the binary variable $x \in \{0, 1\}$.
x, p	Classical kinematic variables.
$\boldsymbol{x}, \boldsymbol{p}$	Quantum physical kinematic operators.
z^*	Complex conjugate of the complex variable $z \in \mathbb{C}$.
ZUPQC	Zero-error uniform polynomial quantum circuit.

Acknowledgments

I thank Prof. Dr. Banzhaf for proposing this project that offered to me the possibility to gain insights into quantum physics and computer science that lectures cannot provide. My gratitude goes to Prof. Dr. Suter who "adopted" me when Prof. Dr. Banzhaf left the University of Dortmund to accede his position as Head of Department of Computer Science at the Memorial University of Newfoundland.

I am much obliged to Prof. Dr. Suter for his help in preparing the experimental realization of the two-qubit parity QC. Also, this experiment wouldn't have been possible without the help of Dipl.-Phys. Hans Georg Krojanski, Dr. Xinhua Peng and Dipl.-Phys. Christian Steffen.

Special thanks go to the members of the Chair of System Analysis at the Department of Computer Science at the University of Dortmund. I'm greatly indebted to Dr. André Leier, Dr. Udo Feldkamp, Dipl.-Inform. Wolfgang Kantschik and Dipl.-Inform. Christian Lasarczyk for all the discussions and help they offered.

I would also thank Rolf Woeste for the numerous discussions about epistemological aspects of physics and mathematics that resulted in insights some of which found their articulation in this thesis.

This work wouldn't have been possible without the financial founding provided by Deutsche Forschungsgemeinschaft (DFG) within the Ph. D. program GK 726 "Materials and Concepts for Quantum Information Processing".

1 Introduction

During the last decade physicists could oversee the rise of increased efforts in investigating foundational problems of non-relativistic quantum physics. Heisenberg's original approach to account for quantum phenomena was to ascribe them to the finiteness of Planck's constant from which it follows that an observation inevitably disturbs the system under investigation [Hei27]. Till today this kind of reasoning can be found in many textbooks.

This approach still left open the possibility that there are elements of the physical reality underlying quantum phenomena. It was realized several decades ago that such "hidden variable theories" contradict the predictions of quantum physics [KS67, Bel87, Mer93]. Nevertheless, experimental evidence has been achieved only recently that quantum physics indeed provide a complete description of the physical world that cannot be completed by reverting to classical physics via hidden variables that are to restore causality and locality to it [WJS+98, ZWJA05].

Furthermore, it was realized very soon [Sch35] that the wave function can be interpreted as a catalog of informations and that physical phenomena are somehow defined by the questions one asks [Boh49]. Nonetheless, it took until recently that theoreticians began to recast quantum physics in terms of information theory that provides a fundamentally new language for studying the relationship between classical and quantum physics [Zei99a, BZ05].

As well as physicists benefit from information theory, computer scientists benefit from quantum physics: The concept of using quantum physical devices to perform computation has received much attention among computer scientists due to the celebrated factoring algorithm proposed by Shor [Sho94]. This algorithm provides an exponential speed-up in comparison to all known classical factoring algorithms. Unfortunately, increased efforts following this promising outcome did not result in any further comparably impressive developments. More recent results have cooled expectations which proved to be too optimistic and it still remains unclear whether quantum computers will provide an alternative superior to classical computation [BV97, BBBV97, BBC+01].

But even modest speed-ups of quantum algorithms (QAs) as compared to classical algorithms might provide further insight into the fundamental differences between classical and quantum physics that can be helpful in estimating the computational benefit of quantum computers.

Unfortunately, the design and development of QAs is a very cumbersome task mainly due to the non-intuitive character of quantum physics. Therefore, it is reasonable to investigate automated algorithm design techniques in the development of new QAs. The usage of Genetic Programming (GP) in performing this task was pioneered by Williams *et al.* [WG98] and Spector *et al.* [SBBS99]. Ever since several related strategies using GP to aid the development of QAs have been proposed and investigated by several authors (e.g., [Rub01, LB03a, MCS04]). Nevertheless, to our knowledge these approaches were only successful in designing quantum circuits

[SBBS99, BBS00, Rub01, MCS04] or optimizing known QAs [LB03a].

The aim of this thesis is to demonstrate that GP also provides a beneficial tool in designing new "better-than-classical" QAs. This goal is supported by the formerly unknown better-than-classical QAs developed by us that will be presented in Chap. 5 and that were published recently [SSB05, SSB].

Most QAs are naturally stated using the so called blackbox model of computation. Therefore we restricted our GP system to evolve quantum decision trees that provide the appropriate devices to keep track of the different results returned by blackbox queries. Nearly every problem can be cast as a decision problem, hence we confined our investigations to this class of problems.

Moreover, the blackbox model of computation provides the only model for which the benefit of quantum computation over classical computation is established. Shor's algorithm [Sho94], for example, reduces the problem of integer factoring to the blackbox problem of order-finding, which is known as being only solved efficiently on quantum computers [Cle99].[1]

This thesis is organized as follows: Chapter 2 provides the mathematical notions and tools of quantum physics and quantum computation used throughout this thesis. It also provides a criterion to distinguish classical from quantum systems. In the end of this chapter we will introduce the main principles of liquid state NMR quantum computers (NMR-QC). The parity algorithms that we developed, were implemented on this device as will be discussed in Sec. 5.3.5. Chapter 3 serves to introduce the main concepts of computability, algorithms and circuits. It also puts the blackbox model of computation into perspective and presents the concept of quantum circuits used by us to represent quantum decision trees. After a short introduction to optimization algorithms our GP system will be discussed in Chap. 4. Finally we will present the QAs developed by us with the help of GP in Chap. 5. There we also will demonstrate the experimental results of implementing the parity algorithm on an NMR-QC.

[1] It may still be possible that integer factoring can be reduced to a problem also efficiently solvable on classical computers.

2 Quantum Physics

In this chapter we will introduce the mathematical formalism of quantum physics used throughout this thesis. Despite its impressive success in describing physical phenomena several authors emphasize that quantum physics still lacks for a foundational principle [Zei99a, Fuc03]. Recently, A. Zeilinger proposed the principle of a "finite information content" of quantum systems that demystifies such controverse topics like the measurement process [Zei99a]. The full impact of this foundational principle is still a topic of current research [BZ05].

Unfortunately, many textbooks [NC00, Bal98, Per95, CTDL96] introduce quantum physics by means of a set of axioms or postulates that are justified by some selected experiments. Duhem criticized such an approach to be questionable from an epistemological point of view [Duh91]. Therefore, we have chosen to motivate the axioms that will be stated in Sec. 2.4 by their historical origin. Nevertheless, this historical oriented approach that will be presented in Sec. 2.1 through Sec. 2.3 can only serve as a heuristic justification.

In Sec. 2.5 we will introduce spinors and Dirac's notation [Dir39] that will be used throughout this thesis. To clarify the differences between classical and quantum physics we will present Bell's inequalities [Bel87] in Sec. 2.6. These inequalities can be used to derive a condition that allows to distinguish classical from quantum systems. Sec. 2.7 generalizes the concept of pure quantum states to mixed quantum states that are necessary to treat NMR-QCs that will finally be presented in Sec. 2.8. In Sec. 2.7 we also will explain why projective measurements and unitary operators are sufficient to treat the dynamics of any quantum computer.

2.1 The Concept of Quanta

The development of quantum physics started with Planck's theoretical derivation [Pla00b] of an interpolated formula [Pla00a] proposed by him to account for the new experimental results on blackbody radiation [RK00]. In this derivation he considered charged linear oscillators that interact with an electromagnetic field. In order to derive his formula he assumed that these oscillators exchange energy E with the electromagnetic field by discrete amounts $E = h\nu$. Here ν denotes the frequency of the electromagnetic field and h denotes the Planck constant. This assumption is in thorough contrast to classical physics where the energy exchange is described by a continuous process. As stated by Pais [Pai79] the only justification for this renunciation of a continuous energy transfer was that it let to the desired answer, namely Planck's formula.

The physical relevance of Planck's ad hoc hypothesis was not realized for several years. As noted by Pais [Pai79] Einstein's struggle to understand the physical concepts underlying Planck's derivation led him to his light-quantum hypothesis [Ein05]. Nevertheless, according to Jammer [Jam66] it was Einstein's successful application

of the concept of quanta to inconsistencies in the classical molecular kinetic theory that endowed Planck's ad hoc hypothesis with physical significance [Ein07a, Ein07b]. In these articles Einstein used the quantization of harmonic oscillators to explain the temperature dependency of the specific heat of bulks. He also showed that the Dulong-Petit rule is reached in the high temperature limit. Einstein's light-quantum hypothesis, on the other hand, was a matter of controversy as it contradicted too much the well established undulatory character of light.

The next successful application of Planck's concept of energy quanta was given by Bohr who used them to explain the line spectra of atoms [Boh13]. To do so he combined the concept of energy quanta with Rutherford's atom model [Rut11] in order to calculate the frequencies emitted or absorbed when the atom passes from one stationary state to another. Bohr's research resulted in an expression for Rydberg's constant of hydrogen that was well confirmed by the experimental data. Bohr's investigation of line spectra also led him to formulate the so called *correspondence principle* [Boh23] that provided a method to quantize a physical system by replacing the differential quotients in its classical description by difference quotients and the *quantization rule* $d\Phi/dn = h$. Here $\Phi = \oint p\,dx$ denotes the action integral, p the kinematic impulse and x the trajectory of the electron in it's nth stationary state.[1]

2.2 Periodic Phenomena and Matrix Mechanics

According to Waerden [Wae68] the research that finally led to quantum physics may be described as systematic guessing, guided by Bohr's principle of correspondence. Unfortunately, systems with several electrons couldn't be treated satisfactorily using this principle. This apparent difficulty led Heisenberg to abandon the classical concept

[1] The derivation of this quantization rule is illustrated in [SW93]: To explain the spectra emitted by an atom Bohr identified the radiation frequency $\nu(n, n-1)$, emitted in the transition from the nth stationary state to the $(n-1)$st stationary state, with the frequency ν_{class} to be expected from the classical theory of radiation for $n \gg 1$. Bohr postulated that the frequency of the emitted light depends on the energy difference of the corresponding stationary states $E(n)$ and $E(n-\alpha)$: $\nu(n, n-\alpha) = [E(n) - E(n-\alpha)] \cdot h^{-1}$. He also showed (see §3 in [Boh13]) that $\nu(n, n-\alpha)$ equals $\alpha\nu_{class}$ for $\alpha \ll n$.

Using these results one obtains a transformation rule from the classical regime to the quantum regime that replaces differential quotients with difference quotients:

$$\nu(n, n-\alpha) = \alpha\,\nu_{class} = \frac{1}{h} \cdot \underbrace{[E(n) - E(n-\alpha)]}_{\text{quantum}} \approx \alpha \cdot \frac{1}{h} \cdot \underbrace{\frac{dE(n)}{dn}}_{\text{classical}} \quad \text{for } \alpha \ll n.$$

This identification returns the condition $\nu_{class} = h^{-1} \cdot dE(n)/dn$ that results in the following quantization rule:

$$\frac{d\Phi}{dn} = h \;\text{ with }\; \Phi = \oint p\,dx \;\text{ and }\; p = \sqrt{2m(E - U(x))}.$$

Here we used that ν_{class} can also be calculated via:

$$\frac{1}{\nu_{class}} = \oint dt = \oint \frac{m}{p}\,dx = \frac{d\Phi}{dE},$$

m denotes the mass of the electron, E it's total and $U(x)$ it's potential energy. The last equality follows due to the definition of the action variable Φ and the kinematic impulse p above. Comparing both expressions for ν_{class} returns the quantization rule presented above, namely $d\Phi/dn = h$.

of trajectories $x(t)$ used to describe the dynamics of electrons [Hei25]. As stated by Jammer [Jam66] Heisenberg hoped to circumvent the tedious guessing work that had to be repeated for each particular quantum-theoretic problem by integrating the correspondence principle in the very mathematical scheme for a new theory of mechanics. Due to the difficulties in assuming classical trajectories he proposed a scheme that replaced the kinematic variables $x(t)$ and $p(t)$ with quantum pendants $\boldsymbol{x}(t)$ and $\boldsymbol{p}(t)$ that were based only on observable magnitudes of the atom's spectra, namely frequency and intensity [Hei25].[2]

This idea of Heisenberg was extended by Born *et al.* to the first conceptual framework of quantum physics which was able to treat periodic phenomena [BJ25, BHJ25]: Defining the quantum version of a classical kinematic variable $a(t)$ by a Hermitian matrix $\boldsymbol{a}(t)$:[3]

$$a(t) \quad \rightarrow \quad (\boldsymbol{a}(t))_{nm} = a(n,m)e^{2\pi\imath\nu(n,m)t} \quad \text{with} \;\; a(n,m) \in \mathbb{C} \;\; \text{and} \;\; \nu(n,m) \in \mathbb{R}, \tag{2.1}$$

any *well ordered* classical function $g(x,p)$ that is given by a power series over the kinematic variables x and p can be transformed into its quantum version by replacing x and p by their quantum pendants $\boldsymbol{x}$ and $\boldsymbol{p}$.[4] Ordinary addition is replaced by matrix addition and ordinary multiplication by matrix multiplication:

$$g(x,p) \quad \rightarrow \quad (\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}))_{nm} = g(n,m)e^{2\pi\imath\nu(n,m)t}.$$

Born *et al.* showed that Hamilton's principle $\int_{t_0}^{t_1} L\,dt = extremal$ translates into extremalizing the trace $tr\{\boldsymbol{L}\}$. Here $\boldsymbol{L}$ denotes the quantum pendant of the classical Lagrangian L defined by $L = p\dot{q} - H(pq)$, with H denoting the Hamilton function. The trace $tr\{\boldsymbol{A}\}$ of an operator $\boldsymbol{A}$ is the sum over its diagonal elements.

[2] The main ideas are presented in [Hei25, FP02, AMS04] and can be summarized as follows: When the electron's periodic motion $x(t)$ is expanded by the Fourier series $x(t) = \sum_\alpha x_\alpha \cdot e^{\imath\alpha\omega t}$ it is possible to associate, by means of the correspondence principle, the terms $\alpha\omega$ with the frequency $\omega(n, n-\alpha)$ emitted by the transition of the electron from the nth to the $(n-\alpha)$th state. The coefficients x_α are related to the intensities of the radiation emitted by this transition. This can be seen by calculating the power $P \sim (\ddot{d}(t))^2$ emitted by the oscillating dipole $d(t) = -e \cdot x(t)$. One gets $P = \sum_\alpha P(n, n-\alpha)$ with $P(n, n-\alpha) \sim |x(n, n-\alpha)|^2$. Here we replaced x_α by the notation $x(n, n-\alpha)$. Thus, Heisenberg concluded that the quantum mechanical pendant $\boldsymbol{x}(t)$ to the classical trajectory $x(t)$ is given by the set $\{x(n, n-\alpha)\, e^{\imath\omega(n,n-\alpha)\cdot t}\}$. From the decomposition of $x(t)$ into a Fourier series he also derived a multiplication rule for the kinematic variables $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$ which turned out to be the rule of matrix multiplication. To conclude this kind of reasoning he finally had to introduce the quantization rule into his new formalism. Born *et al.* showed that the quantization rule derived by Heisenberg can be replaced by Eq. (2.3) [BJ25].

[3] As demonstrated in footnote 2 the representation of a quantum theoretical variable $\boldsymbol{a}(t)$ by a matrix was suggested by Heisenberg due to the fact that the frequencies $\nu(n,m)$ and coefficients $a(n,m)$ in the Fourier expansion of a classical periodic variable $a_n(t) = \sum_m a(n,m)e^{2\pi\imath\nu(n,m)t}$ can be interpreted in terms of observable quantities. As $a_n(t)$ has to be real the conditions $a(n,m) = a^*(m,n)$ and $\nu(n,m) = -\nu(m,n)$ have to be fulfilled. Thus, it follows that $\boldsymbol{a}$, as defined in Eq.(2.1), has to be Hermitian.

[4] The matrices $\boldsymbol{x}$ and $\boldsymbol{p}$ do not commute. For $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p})$ to be Hermitian the function $g(x,p)$ has to be transformed into a well ordered function so that the succession of x's and p's is symmetric:

$$g(x,p) = xp^2 \quad \rightarrow \quad g(x) = \frac{xp^2 + p^2x}{2}.$$

This well ordered function can be transformed into its quantum pendant $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p})$ that now is Hermitian by construction.

The authors derived the canonical equations of motion:[5]

$$\dot{\boldsymbol{p}} = -\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{x}}, \quad \dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{p}}. \tag{2.2}$$

Then they demonstrated that Bohr's quantization rule $d\Phi/dn = h$ translates into:

$$\boldsymbol{x}(t) \cdot \boldsymbol{p}(t) - \boldsymbol{p}(t) \cdot \boldsymbol{x}(t) = \frac{\hbar}{\imath}\mathbf{1}, \text{ with } \hbar = \frac{h}{2\pi}, \tag{2.3}$$

here $\mathbf{1}$ denotes the identity matrix.

It follows that a quantum theoretical problem is solved when the solutions of the canonical equations of motion in Eq. (2.2) fulfill the quantization condition of Eq. (2.3).

Finally the authors showed that this problem is equivalent to the eigenvalue problem:

$$\left(\boldsymbol{U}^{-1}\boldsymbol{H}\boldsymbol{U}\right)_{nm} = E(n)\delta_{nm}, \tag{2.4}$$

where $\boldsymbol{U}$ denotes a transformation that leaves Eq. (2.3) invariant.

2.3 Quantization and Sturm-Liouville Eigenvalue Problems

The following remark of E.U. Condon [Con62] is well suited to illustrate the close relationship of Heisenberg's matrix mechanics and Schrödinger's wave mechanics that will be discussed thereafter:

> *"[...] Hilbert was having a great laugh on Born and Heisenberg and the Göttingen theoretical physicists because when they first discovered matrix mechanics they were having, of course, the same kind of trouble that everybody else had in trying to solve problems and to manipulate and really do things with matrices. So they went to Hilbert for help, and Hilbert said the only times that he had ever had anything to do with matrices was when they came up as a sort of by-product of the eigenvalues of the boundary-value problem of a differential equation. So if you look for the differential equation which has these matrices you can probably do more with that. They had thought it was a goofy idea and that Hilbert did not know what he was talking about, so he was having a lot of fun pointing out to them that they could have discovered Schrödinger's wave mechanics six months earlier if they had paid a little more attention to him."*

Schrödinger derived his equation from a completely different point of view than Heisenberg [Sch26a, Sch26b]. He further showed that the eigenvalue problem in Eq.

[5] The time derivative of $(\boldsymbol{a})_{nm} = a(n,m)e^{2\pi\imath\nu(n,m)t}$ is defined by $(\dot{\boldsymbol{a}})_{nm} = 2\pi\imath\nu(n,m)\,(\boldsymbol{a})_{nm}$. The derivative $\partial_{\boldsymbol{a}}\boldsymbol{g}(\boldsymbol{a})$ is defined by:

$$\frac{\partial \boldsymbol{g}(\boldsymbol{a})}{\partial \boldsymbol{a}} = \lim_{\alpha \to 0} \frac{\boldsymbol{g}(\boldsymbol{a} + \alpha\mathbf{1}) - \boldsymbol{g}(\boldsymbol{a})}{\alpha}.$$

(2.4) is equivalent to solve his Sturm-Liouville eigenvalue problem nowadays known as Schrödinger's equation [Sch26c]:

$$\boldsymbol{H}\psi(x) = -E\psi(x). \tag{2.5}$$

Here $\boldsymbol{H}$ denotes the operator one gets by replacing the classical kinematic variables x and p in the well ordered Hamilton function $H(x,p)$ by the operators $\boldsymbol{x} = x$ and $\boldsymbol{p} = -\imath\hbar\partial_x$; $\psi(x)$ denotes a single valued square integrable function over the configuration space.

The problem to calculate the energy E_n of the electron's nth stationary state now translates into the pure mathematical problem to find the eigenfunctions u_n of the following eigenvalue problem:

$$\boldsymbol{H}u_n(x) = -E_n u_n(x). \tag{2.6}$$

With Schrödinger's approach Heisenberg's quantization rule is automatically fulfilled.

A scalar product $\langle u_i, u_j\rangle$ between two square integrable functions u_i and u_j can be defined by:

$$\langle u_i(x), u_j(x)\rangle = \int_x u_i^*(x)u_j(x)\,dx.$$

It is well known from the Sturm-Liouville theory that a Hermitian operator $\boldsymbol{H}$ of the eigenvalue problem $\boldsymbol{H}u_i = \lambda_i u_i$ has real eigenvalues λ_i and orthogonal eigenfunctions u_i [Jän01].[6] If $\boldsymbol{H}$ is a linear operator of second order then the set of eigenfunctions is also complete [Wei]. In this case any square integrable function ψ can be written as a linear combination over the eigenfunctions u_i of $\boldsymbol{H}$:

$$\psi(x) = \sum_i c_i u_i(x) \quad \text{with} \quad c_i = \langle \psi, u_i\rangle. \tag{2.7}$$

As $\boldsymbol{x}$ and $\boldsymbol{p}$ are Hermitian operators the operator $\boldsymbol{H}(\boldsymbol{x},\boldsymbol{p})$ also is Hermitian when it is well ordered in $\boldsymbol{x}$ and $\boldsymbol{p}$. The eigenfunctions u_i can be normalized, therefore it is possible to find a set of eigenfunctions that fulfill the orthonormality condition defined by $\langle u_i, u_j\rangle = \delta_{ij}$. Due to Eq. (2.6) the eigenvalues E_i can be calculated by:

$$E_i = \langle u_i, \boldsymbol{H}u_i\rangle. \tag{2.8}$$

Despite the non-intuitive character of Schrödinger's equation it provides the advantage to also describe non-periodic phenomena. Also, it is possible to treat non-stationary phenomena by replacing the energy term E in Eq. (2.5) by $E \to -\imath\hbar\partial_t$ as proposed by Schrödinger [Sch26d]:

$$\boldsymbol{H}\psi(x,t) = \imath\hbar\partial_t\psi(x,t). \tag{2.9}$$

Schrödinger's attempts to attribute a physical significance to his wave function ψ did not return satisfactory results. They were replaced by the statistical interpretation proposed by Born [Bor26]. He interpreted the absolute square $|\psi|^2$ of Schrödinger's

[6] An operator $\boldsymbol{H}$ is called Hermitian if it fulfills the following condition [Jän01]:

$$\langle u_i, \boldsymbol{H}u_j\rangle = \langle \boldsymbol{H}u_i, u_j\rangle.$$

wave function to represent a probability density. This proposal was based on Born's results he had obtained by investigating scattering processes. Thus, the expectation value $\langle x \rangle$ of the particle position x is calculated by:

$$\langle x \rangle(t) = \int_x x|\psi(x,t)|^2 \, dx = \int_x \psi^*(x,t) \, \boldsymbol{x} \, \psi(x,t) \, dx = \langle \psi, \boldsymbol{x}\psi \rangle.$$

It follows that the expectation value $\langle p \rangle$ of the particle's impulse p can be calculated by [Deh00]:

$$\begin{aligned} \langle p \rangle(t) = m\frac{d\langle x \rangle(t)}{dt} &= \int_x mx \left(\psi^*(x,t)\partial_t \psi(x,t) + (\partial_t \psi^*(x,t)) \, \psi(x,t) \right) dx \\ &= \int_x \psi^*(x,t) \left(-\imath\hbar\partial_x \right) \psi(x,t) \, dx = \langle \psi, \boldsymbol{p}\psi \rangle. \end{aligned}$$

The last but one equation follows from inserting Schrödinger's equation for the terms $\partial_t \psi$ and from using that ψ is square integrable.

These results show that the expectation value $\langle g \rangle$ of any well ordered function $g(x,p)$ that can be written as a series expansion over x and p is calculated via:

$$\langle g \rangle = \langle \psi, \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p})\psi \rangle.$$

Replacing g by the Hamiltonian H and using Eq. (2.7) and Eq. (2.8) one gets:

$$\langle H \rangle = \langle \psi, \boldsymbol{H}\psi \rangle = \sum_{i,j} c_i^* c_j \langle u_i, \boldsymbol{H} u_j \rangle = \sum_i c_i^* c_i E_i = \sum_i p_i E_i.$$

We introduced p_i to denote the probability that a measurement returns the eigenvalue E_i.

2.3.1 Composite Systems

Consider a composite physical system $S^{(AB)}$ described by the Hamiltonian $H^{(AB)}$:

$$H^{(AB)}(x_1, x_2, p_1, p_2) = H^{(A)}(x_1, p_1) + H^{(B)}(x_2, p_2) + V(x_1, x_2),$$

where $H^{(A)}$ describes the Hamiltonian of subsystem $S^{(A)}$, $H^{(B)}$ the Hamiltonian of subsystem $S^{(B)}$ and V an interaction between these two subsystems. If the corresponding Schrödinger equation is separable then the completeness of the individual system's eigenfunctions $u_i^{(A)}$ and $u_i^{(B)}$ that are defined by:

$$\boldsymbol{H}^{(A)} u_i^{(A)} = E_i^{(A)} u_i^{(A)} \quad \text{and} \quad \boldsymbol{H}^{(B)} u_i^{(B)} = E_i^{(B)} u_i^{(B)},$$

effects that any solution of $\boldsymbol{H}^{(AB)} \psi^{(AB)}(x_1, x_2) = E^{(AB)} \psi^{(AB)}(x_1, x_2)$ can be written as:

$$\psi^{(AB)}(x_1, x_2) = \sum_l c^l u_l^{(AB)}(x_1, x_2) = \sum_{l,i,j} \tilde{c}_{ij}^l u_i^{(A)}(x_1) \cdot u_j^{(B)}(x_2), \quad c^l, \tilde{c}_{ij}^l \in \mathbb{C}.$$

Therefore, the vector space of the system's eigenfunctions $u_l^{(AB)}$ is spanned by the tensor product, denoted by $\otimes$, of the subsystem's eigenfunctions $u_i^{(A)}$ and $u_j^{(B)}$:

$$u_l^{(AB)}(x_1, x_2) = \sum_{i,j} c_{ij}^l u_i^{(A)}(x_1) \otimes u_j^{(B)}(x_2) = \sum_{i,j} c_{ij}^l u_i^{(A)}(x_1) \cdot u_j^{(B)}(x_2), \quad c_{ij}^l \in \mathbb{C}.$$

2.4 Axiomatic Approach to Quantum Physics

The last results suggest the following mathematical structure of quantum physics that is usually stated in the form of axioms (see for example [CTDL96]):

1. Any physical state is represented by a wavefunction ψ that can be decomposed into a superposition $\psi = \sum_i c_i u_i$ over the eigenstates u_i of a Hermitian operator $\boldsymbol{A}$.[7] This operator, also called observable, is associated with a measurable property, e.g. the energy E, of the physical system. The probability p_i to measure the eigenvalue a_i of $\boldsymbol{A}$ is $p_i = c_i^* c_i$. The vector space spanned by the eigenvectors u_i forms an inner product space. It is complete with respect to the norm defined by the scalar product (inner product). Such spaces are called *Hilbert spaces*.

2. The time evolution of the physical state ψ is governed by Schrödinger's equation.

3. Quantum states of the composite system S^{AB} that consists of the subsystems S^A and S^B are vectors of the Hilbert space $\mathcal{H}^{AB}$ which is the tensor product of the individual subsystems' Hilbert spaces $\mathcal{H}^A$ and $\mathcal{H}^B$: $\mathcal{H}^{AB} = \mathcal{H}^A \otimes \mathcal{H}^B$.

Systems that are prepared into the eigenstate u_i of the observable $\boldsymbol{A}$ return the same measurement result a_i. Because non-commuting Hermitian operators $\boldsymbol{A}$ and $\boldsymbol{B}$ do not share the same set of eigenfunctions, measurements of the observable $\boldsymbol{B}$ will not return a unique value.

If one insists on the wavefunction ψ to provide a complete description of the physical phenomenon then a measurement, in order to have an operational meaning, has to describe a procedure that prepares the system: Once a non-destructive measurement of the observable $\boldsymbol{B}$ returns the measurement result b_i, all following measurements of $\boldsymbol{B}$ also have to return this value.[8] Hence, additional to the quantum state's time evolution postulated by Axiom 2, one introduces a collapse process:

4. A quantum state ψ described by a superposition over the eigenstates u_i of the Hermitian operator $\boldsymbol{A}$:

$$\psi = \sum_i c_i u_i \quad \text{with} \quad u_i : \ \boldsymbol{A}\, u_i = a_i u_i \ \text{ and } \ c_i \in \mathbb{C}, a_i \in \mathbb{R},$$

is mapped to the state:

$$\psi = v_j \quad \text{with} \quad v_j : \ \boldsymbol{B}\, v_j = b_j v_j \ \text{ and } \ b_j \in \mathbb{R},$$

upon measurement of the observable's $\boldsymbol{B}$ eigenvalue b_j. Such a measurement projects the state ψ to v_j. Measurements of this character are called *projective measurements*.

[7] A Sturm-Liouville eigenvalue problem has an infinite dimensional Hilbert space. Additional to hermiticity one demands that the Liouville operator is of second order. Otherwise, the space spanned by the eigenvectors might not be complete [Jän01]. For finite dimensional vector spaces hermiticity of an operator is sufficient to provide a complete set of eigenvectors [Fis89].

[8] Here we assume that the time interval between two successive measurements is short enough so that the time evolution does not noticeably change the state.

2.5 Spinors and Dirac's Notation

Born's probability interpretation led to severe conceptual problems in the interpretation of quantum physics that were initially raised by Einstein [Boh49]. His concerns where motivated by a thought experiment where particles are diffracted at a single slit. The detection of such particles on the detection screen is a spatially confined process whereas in quantum physics the particle is described by a spatially extended wavefunction. Born's probability interpretation, according to Einstein, implies an instantaneous collapse of the wavefunction that prevents the wavefunction to act simultaneously on two different points of the detection screen.

According to Einstein such a collapse contradicts the conception of locality as the information that causes the detection at a single point has to be distributed throughout the wavefront instantaneously which contradicts relativity.

Here the different conceptions of reality in classical physics on the one hand and quantum physics on the other become apparent. On one side the proponents of quantum physics insist that the wave function provides a complete description of the physical phenomenon and that it makes no sense to attribute a position to a particle until it hits the detection screen. On the other side, motivated by the classical conception of reality, Einstein tried to show that the quantum physical description provided by Schrödinger's equation and Born's probability interpretation does not offer a complete description of the physical phenomena. One of the most influential contributions of Einstein to settle the question of completeness of quantum physics is the so called EPR paper [EPR35] that indicates a way to address these questions experimentally as elaborated by Bell [Bel87] (see Sec. 2.6).

EPR's and Bell's argument utilizes states of a compound system that consists of two spatially separated subsystems. These states are of a special form called entangled. Due to the prominent status of entanglement in quantum computation we will present a short characterization of such states. Before doing so we have to introduce, additional to the external states $\psi(x)$, internal spinor states of spin-1/2 particles. External as well as internal states form vector spaces. We also introduce Dirac's notation that provides the same representation of quantum states irrespective of being external or internal. Spin-1/2 particles constitute a prototype for two-state systems that represent an appropriate quantum version of classical bits. Furthermore, spin-1/2 particles will be of interest when we present NMR-QCs in Sec. 2.8. There the molecule's spin-1/2 nuclei serve as quantum bits also called *qubits*.

From now on, the underscore in $\underline{v}$ denotes that $\underline{v}$ is a vector. The bold font in $\boldsymbol{v}$ indicates that $\boldsymbol{v}$ is an operator. Hence, $\underline{\boldsymbol{v}}$ denotes a vector of operators.

The coupling of a free spin-1/2 particle to the electromagnetic field is described by Pauli's equation:

$$-\left(\frac{\hbar^2}{2m}\underline{\partial_x}^2+\frac{e\hbar}{2mc}\underline{\boldsymbol{\sigma}}\cdot\underline{B}\right)\cdot\underline{\varphi}(\underline{x},t)=\imath\hbar\partial_t\underline{\varphi}(\underline{x},t), \tag{2.10}$$

where $\underline{B}$ denotes the magnetic field vector, e the particle's charge, $\underline{x}$ the particle's position, $\underline{\partial}_x$ the gradient operator and $\underline{\partial_x}^2$ the Laplace operator $\partial_x^2+\partial_y^2+\partial_z^2$.[9] The

[9] As Schrödinger's equation is not invariant under Lorentz transformation one replaces it with Klein-Gordon's equation:

$$-\hbar^2\partial_t^2\psi=\left(-\hbar^2c^2\underline{\partial_x}^2+m^2c^4\right)\psi,$$

Pauli matrices $\boldsymbol{\sigma}_x$, $\boldsymbol{\sigma}_y$, $\boldsymbol{\sigma}_z$ and the identity operator $\mathbf{1}$ are given by:

$$\boldsymbol{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_y = \begin{pmatrix} 0 & -\imath \\ \imath & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The state $\underline{\varphi}$ has the two components φ_1 and φ_2, both are single valued square integrable functions:[10]

$$\underline{\varphi} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix}.$$

Analogous to the external states ψ the internal states, described by the solutions $\underline{\varphi}$ of Pauli's equation, form a vector space. Furthermore, it is possible to define a scalar product $\langle \underline{\zeta}, \underline{\varphi} \rangle$ for two solutions $\underline{\varphi}$ and $\underline{\zeta}$ via:

$$\langle \underline{\zeta}, \underline{\varphi} \rangle = (\zeta_1^*, \zeta_2^*) \cdot \begin{pmatrix} \varphi_1 \\ \varphi_2 \end{pmatrix} = \zeta_1^* \varphi_1 + \zeta_2^* \varphi_2.$$

Due to this common vector space structure of external and internal quantum states Dirac introduced an alternative notation to facilitate calculations [Dir39]. Applying this notation the wavefunction ψ is denoted by $|\psi\rangle$. By introducing the dual $\langle\psi|$ of the vector $|\psi\rangle$ the scalar product of the two states $|\phi\rangle$ and $|\psi\rangle$ is denoted by $\langle\phi|\psi\rangle$. According to Born's probability interpretation the absolute square $|\psi|^2 = \langle\psi|\psi\rangle$ of the wavefunction $|\psi\rangle$ must be normalized to one: $|\psi|^2 = 1$.

The dyadic product $|\psi\rangle\langle\phi|$ denotes a linear operator that maps $|\psi\rangle$ onto $|\phi\rangle$ via the operation:

$$|\psi\rangle\langle\phi| \cdot |\psi\rangle = |\psi\rangle \cdot \langle\phi|\psi\rangle.$$

Using Dirac's notation Schrödinger's equation now looks as follows:

$$\boldsymbol{H}|\psi\rangle = \imath\hbar\partial_t|\psi\rangle. \tag{2.11}$$

that follows from the relativistic energy-impulse dependency $E^2 = \underline{p}^2c^2 + m^2c^4$ applying the canonical quantization:

$$\underline{p} \to \frac{\hbar}{\imath}\underline{\partial_x}, \quad E \to \imath\hbar\partial_t.$$

According to Axiom 3 a measurement causes a collapse of the wavefunction. Thus, the occurrence of a second time derivative in the Klein-Gordon equation gives rise to problems as the two initial values $\psi(0)$ and $\dot{\psi}(0)$ cannot be provided by a measurement. To circumvent this problem Dirac proposed the following equation [Dir39]:

$$\imath\hbar\partial_t\psi = \left(\frac{\hbar c}{\imath}\underline{\boldsymbol{\alpha}}\underline{\partial_x} + \boldsymbol{\beta}mc^2\right)\psi,$$

also called Dirac equation. As the wave function ψ also has to fulfill the Klein-Gordon equation the coefficients $\underline{\boldsymbol{\alpha}}$ and $\boldsymbol{\beta}$ have to be matrices of rank of at least 4. Hence, the wavefunction ψ has 4 components the last two of which are attributed to the anti-particle:

$$\underline{\boldsymbol{\alpha}} = \left(\begin{pmatrix} 0 & \boldsymbol{\sigma}_x \\ \boldsymbol{\sigma}_x & 0 \end{pmatrix}, \begin{pmatrix} 0 & \boldsymbol{\sigma}_y \\ \boldsymbol{\sigma}_y & 0 \end{pmatrix}, \begin{pmatrix} 0 & \boldsymbol{\sigma}_z \\ \boldsymbol{\sigma}_z & 0 \end{pmatrix}\right)^t, \quad \boldsymbol{\beta} = \begin{pmatrix} \mathbf{1} & 0 \\ 0 & -\mathbf{1} \end{pmatrix}.$$

To describe the coupling of the particle to the electromagnetic field the kinematic impulse $\underline{p}$ is replaced by the canonical impulse $\underline{p} - (e/c)\underline{A}$, where $\underline{A}$ denotes the vector potential of the electromagnetic field. The non-relativistic case of Dirac's equation is Pauli's equation [Sch05].

[10] For the sake of readability we do not explicitly display time and position dependency of the state $\varphi(x,t)$ from now on.

Due to linearity of Schrödinger's equation the time evolution of the state $|\psi\rangle$ can be described by a linear operator $\boldsymbol{U}(t)$:

$$|\psi(t)\rangle = \boldsymbol{U}(t)|\psi(0)\rangle.$$

If the Hamiltonian $\boldsymbol{H}$ is time independent then the integration of Schrödinger's equation Eq. (2.11) returns:

$$\boldsymbol{U}(t) = e^{-\frac{\imath}{\hbar}\boldsymbol{H}t}. \tag{2.12}$$

Born's probability interpretation of $|\psi(t)|^2 = \langle\psi(t)|\psi(t)\rangle$ demands the linear operators $\boldsymbol{U}$ to be unitary as only unitary transformations preserve the norm [Hir01]:[11]

$$\| \, |\psi(t)\rangle \, \| = \| \, \boldsymbol{U}(t)|\psi(0)\rangle \, \| = \| \, |\psi(0)\rangle \, \| \;\text{ with }\; \|\psi\| = \sqrt{\langle\psi|\psi\rangle}.$$

The dynamics of a system that consists of the two non-interacting spin-1/2 particles A and B is described by:

$$\boldsymbol{H}_A|\psi\rangle + \boldsymbol{H}_B|\psi\rangle = \imath\hbar\partial_t|\psi\rangle. \tag{2.13}$$

It can be solved by the tensor product $|\psi_A\rangle \otimes |\psi_B\rangle$ of the subsystems solutions $|\psi_A\rangle$ and $|\psi_B\rangle$ with:

$$\boldsymbol{H}_A|\psi_A\rangle = \imath\hbar\partial_t|\psi_A\rangle,$$

and

$$\boldsymbol{H}_B|\psi_B\rangle = \imath\hbar\partial_t|\psi_B\rangle.$$

The tensor product of the vectors $|\psi_A\rangle$ and $|\psi_B\rangle$:

$$|\psi_A\rangle = \alpha_A|0\rangle + \beta_A|1\rangle = \begin{pmatrix} \alpha_A \\ \beta_A \end{pmatrix} \quad \text{and} \quad |\psi_B\rangle = \alpha_B|0\rangle + \beta_B|1\rangle = \begin{pmatrix} \alpha_B \\ \beta_B \end{pmatrix},$$

now looks as follows:

$$\begin{aligned} |\psi\rangle &= |\psi_A\psi_B\rangle = |\psi_A\rangle|\psi_B\rangle = |\psi_A\rangle \otimes |\psi_B\rangle = \begin{pmatrix} \alpha_A \\ \beta_A \end{pmatrix} \otimes \begin{pmatrix} \alpha_B \\ \beta_B \end{pmatrix} = \begin{pmatrix} \alpha_A\alpha_B \\ \alpha_A\beta_B \\ \beta_A\alpha_B \\ \beta_A\beta_B \end{pmatrix} \\ &= \alpha_A\alpha_B|00\rangle + \alpha_A\beta_B|01\rangle + \beta_A\alpha_B|10\rangle + \beta_A\beta_B|11\rangle. \end{aligned}$$

Above we introduced $|0\rangle$ and $|1\rangle$ to denote the two orthonormal states of a two-state system, also called computational basis in the context of quantum information processing.

The string 10 in the state $|10\rangle$ above can be interpreted as the binary representation of the integer 2. From now on we use $|\underline{2}\rangle$ to denote the state $|10\rangle$. Thus, the state $|\psi_A\psi_B\rangle$ can be written as:

$$|\psi_A\psi_B\rangle = \sum_{i=0}^{3} c_i|\underline{i}\rangle,$$

with $c_0 = \alpha_A\alpha_B$, $c_1 = \alpha_A\beta_B$, $c_2 = \beta_A\alpha_B$ and $c_3 = \beta_A\beta_B$.

According to Axiom 3 any state of the Hilbert space $\mathcal{H}^{AB}$ spanned by $|\underline{0}\rangle$, $|\underline{1}\rangle$, $|\underline{2}\rangle$ and $|\underline{3}\rangle$ provides a valid solution of Eq. (2.13). This gives rise to so called entangled states that will be discussed in the next section.

[11] A unitary operator $\boldsymbol{U}$ is characterized by the condition $\boldsymbol{U}^{-1} = \boldsymbol{U}^{+}$ where $\boldsymbol{U}^{+}$ denotes the adjoint of $\boldsymbol{U}$: $(\boldsymbol{U}^{+})_{ij} = (\boldsymbol{U}_{ji})^{*}$.

2.6 Entanglement, Hidden Variables and Bell's Inequalities

If one simulates quantum computers on classical computers then one is faced with the problem that the dimension d of the Hilbert space $\mathcal{H}_d$ grows exponentially in the number n of qubits:

$$\mathcal{H}_{2^n} = \underbrace{\mathcal{H}_2 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_2}_{n \text{ times}}.$$

Here $\mathcal{H}_2$ denotes the two-dimensional state space of a single qubit. Therefore, one might be tempted to search for another description of an n-qubit state that reduces the simulation effort.

Another problem is to decide whether a QA really is quantum. One needs a criteria to decide if a QA cannot be implemented as efficiently on a system already describable in terms of classical physics.

Such questions are closely related to "hidden variable theories" and Bell's theorem as will be shown below [Bel87].

Consider for example the following quantum state $|\psi\rangle$ of a two-qubit system that cannot be written as a tensor product of the individual qubits' states:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right) \neq \left(\alpha_1|0\rangle + \beta_1|1\rangle\right) \otimes \left(\alpha_0|0\rangle + \beta_0|1\rangle\right) \quad \text{with} \quad \alpha_0, \alpha_1, \beta_0, \beta_1 \in \mathbb{C}.$$

Such states are called entangled. Due to the measurement postulate (Axiom 4 in Sec. 2.4) a measurement on one qubit immediately determines the measurement result on the other.

From now on we assume that the two entangled qubits are realized by the spins' z-components of two spin-1/2 particles. As mentioned before, the measurement of the z-component of one spin immediately determines the result of a corresponding measurement on the second spin. This correlation in the measurement results suggests that the qubits' spins were already determined in the very beginning of the whole experiment. Nevertheless, according to quantum physics, the wave function $|\psi\rangle$ provides a complete description of the system and must not be extended. As shown by Bell [Bel87] this problem is closely related to the EPR paradox [EPR35] where an argument was presented that quantum physics had to be supplemented by additional variables to restore causality and locality.

In hidden variable theories correlations between spin measurements are also possible due to the finiteness of Planck's constant. A measurement of one particle's spin alters the whole system and thus affects the measurement result for the other. One has to separate the two particles by a sufficiently large spatial distance. According to the locality assumption a measurement on one particle cannot instantaneously influence the measurement on the other. Now correlations in the measurement results cannot be explained by a mutual classical influence anymore. The assumptions underlying Bell's hidden variables model can therefore be paraphrased by the term "local realism".

All classical physical theories like relativity theory, electromagnetism etc. are realistic theories. Due to the success of relativity theory physicists now also believe that localism is an essential property of the physical world. Hence, the results obtained by Bell's considerations provide us with a criteria to check whether a QA indeed is quantum.

In Fig. 2.1 the scenario of two spin-1/2 particles described above is shown. The measurement device on the left side (particle 1) is aligned along axis $\underline{n}_1$, the measurement device on the right side (particle 2) along $\underline{n}_2$. Quantum physics predicts that a measurement of the observable $\underline{\boldsymbol{\sigma}} \cdot \underline{n}_i$, with $i \in \{0, 1\}$, returns ± 1. Therefore, the result $A(\underline{n}_1)$ of measuring $\underline{\boldsymbol{\sigma}} \cdot \underline{n}_1$ is represented in a hidden variable theory by $A(\underline{n}_1, \lambda) = \pm 1$ where λ denotes the hidden variable. The same holds for the measurement result $B(\underline{n}_2)$ which now is determined by $B(\underline{n}_2, \lambda) = \pm 1$. Provided with the measurement results $A(\underline{n}_1)$ and $B(\underline{n}_2)$ one calculates the correlation function $P(\underline{n}_1, \underline{n}_2)$:

$$P(\underline{n}_1, \underline{n}_2) = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} A_i(\underline{n}_1) B_i(\underline{n}_2),$$

here N denotes the number of measurements. A quantum physical treatment returns:

$$P^{quant}(\underline{n}_1, \underline{n}_2) = \langle \psi | \underline{\boldsymbol{\sigma}} \cdot \underline{n}_1 \otimes \underline{\boldsymbol{\sigma}} \cdot \underline{n}_2 | \psi \rangle.$$

The hidden variables treatment returns:

$$P^{class}(\underline{n}_1, \underline{n}_2) = \int A_\lambda(\underline{n}_1) B_\lambda(\underline{n}_2) \varrho(\lambda) \, d\lambda.$$

Here $\varrho(\lambda)$ denotes the probability distribution of the hidden variable λ. As shown by Bell one finally gets [Bel87]:

$$|P^{class}(\underline{n}_1, \underline{n}_2) - P^{class}(\underline{n}_1, \underline{n}_3)| - P^{class}(\underline{n}_2, \underline{n}_3) - 1 \leq 0,$$

which is known as Bell's inequality. Bell demonstrated that there are directions $\underline{n}_1$, $\underline{n}_2$ and $\underline{n}_3$ with:

$$|P^{quant}(\underline{n}_1, \underline{n}_2) - P^{quant}(\underline{n}_1, \underline{n}_3)| - P^{quant}(\underline{n}_2, \underline{n}_3) - 1 = \frac{1}{2}.$$

Hence, quantum physics and hidden variable theories differ in their predictions. Therefore, the question whether quantum physics is to be supplemented by hidden variables can be treated experimentally. Such experiments were performed and, despite the fact that some loopholes still remain open, contradict the predictions of hidden variable theories based on local realism (see references in [ZWJA05, Zei99b]).

Another version of Bell's inequalities often used in the literature are the so called CHSH inequalities [CHSH69]:

$$|P^{class}(\underline{n}_1, \underline{n}_3) + P^{class}(\underline{n}_1, \underline{n}_4) + P^{class}(\underline{n}_2, \underline{n}_3) - P^{class}(\underline{n}_2, \underline{n}_4)| \leq 2,$$

here measurements on each particle can be chosen between two arbitrary dichotomic observables, namely $\underline{n}_1 \cdot \underline{\boldsymbol{\sigma}}$ or $\underline{n}_2 \cdot \underline{\boldsymbol{\sigma}}$ on particle 1 and $\underline{n}_3 \cdot \underline{\boldsymbol{\sigma}}$ or $\underline{n}_4 \cdot \underline{\boldsymbol{\sigma}}$ on particle 2. The Gisin theorem states that any non-product state of two particles violates a CHSH inequality [Gis91]. Zukowski *et al.* generalized Bell's inequalities to an arbitrary number of qubits [ZB02]. Analogous to the CHSH inequalities, measurements on each particle can be chosen between two arbitrary dichotomic variables. Unfortunately, the Gisin theorem cannot be generalized to these N-particle Bell inequalities as

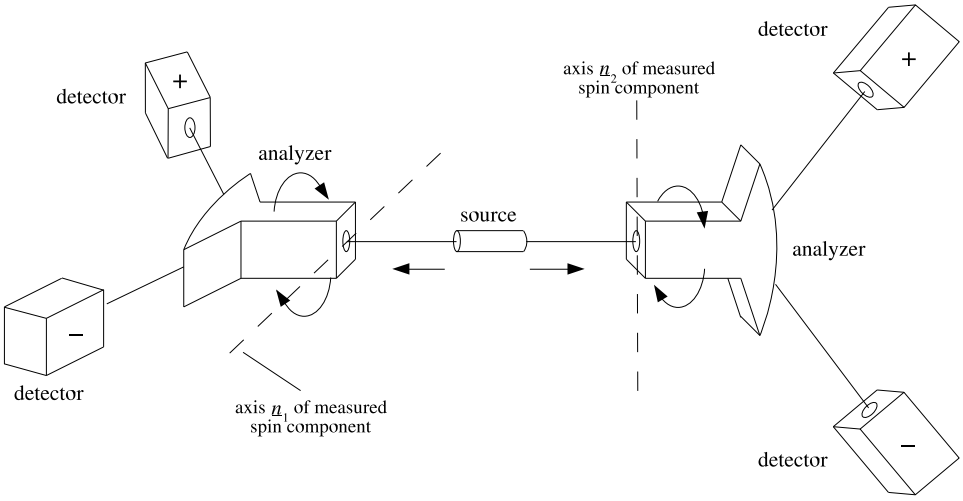

Figure 2.1: Thought experiment to test Bell's inequalities by measuring the spins' components of electrons along the axis $\underline{n}_1$, $\underline{n}_2$, respectively. Along any arbitrary axis a spin's component can only have two possible values denoted by plus and minus. The source creates an entangled pair of electrons, the electrons fly apart in opposite directions.

there exist non-product states that do not violate any of these inequalities [ZBLW02]. Nevertheless, these inequalities are satisfied by any product state.[12]

These results show that any product state can be described classically. The parameter space of product states grows linearly in the number of qubits. The experimental results for correlation functions on 2 and 3 qubits indicate that a linear growth of the parameter space is not sufficient to account for the occurring non-classical correlations [WJS+98, PBD+00]. Therefore, the exponential growth of the parameter space in the number of qubits seems to be a necessary condition to ensure quantumness.

2.7 Mixed States, POVMs and Quantum Operations

Imagine an experimentator whose apparatus prepares a quantum system to be in the state $|\psi_i\rangle \in \mathcal{H}$ with probability p_i. The expectation value $\langle \boldsymbol{A} \rangle$ of the observable $\boldsymbol{A}$ is calculated via:

$$\begin{aligned} \langle \boldsymbol{A} \rangle &= \sum_i p_i \langle \psi_i | \boldsymbol{A} | \psi_i \rangle = \sum_i p_i \, tr \left\{ |\psi_i\rangle\langle\psi_i| \boldsymbol{A} \right\} = tr \left\{ \left(\sum_i p_i |\psi_i\rangle\langle\psi_i| \right) \boldsymbol{A} \right\} \\ &= tr \left\{ \boldsymbol{\varrho} \boldsymbol{A} \right\}, \quad \text{with } \boldsymbol{\varrho} = \sum_i p_i |\psi_i\rangle\langle\psi_i|, \end{aligned} \tag{2.14}$$

[12] From any state that violates any of these inequalities pure state entanglement is distillable [ASW02]. Distillibility denotes a procedure where two separated observers can prepare n entangled states from $m \geq n$ quantum states by only local operations and classical communication. Pure entangled states cannot be distilled from pure product states. Hence, a pure product state satisfies the N-particle Bell inequalities.

here we used the equivalence between the trace $tr\{|\psi_i\rangle\langle\psi_i|\boldsymbol{A}\}$ of the dyadic product $|\psi_i\rangle\langle\psi_i|$ times the observable $\boldsymbol{A}$ and the expectation value $\langle\psi_i|\boldsymbol{A}|\psi_i\rangle$ of the observable $\boldsymbol{A}$. The operator $\boldsymbol{\varrho}$ is usually called *density matrix* and has the following properties [NC00]:

1. Trace condition I: $tr\{\boldsymbol{\varrho}\} = 1$
2. Trace condition II: $tr\{\boldsymbol{\varrho}^2\} \leq 1$, with equality if $\boldsymbol{\varrho} = |\psi\rangle\langle\psi|$, where $|\psi\rangle \in \mathcal{H}$.
3. Positivity condition: $\langle\psi|\boldsymbol{\varrho}|\psi\rangle \geq 0$ for any state $|\psi\rangle \in \mathcal{H}$. From this condition it follows that $\boldsymbol{\varrho}$ also is Hermitian and thus diagonalizable with real eigenvalues.

The density matrix $\boldsymbol{\varrho}$ describes quantum systems from which one only knows the probability p_i of the system to be in the state $|\psi_i\rangle$. Therefore, it provides a more general description of a quantum system than the description based on *pure states* $|\psi_i\rangle \in \mathcal{H}$ alone. Trace condition II provides a criterion that can be used to check if a quantum system, described by the density matrix $\boldsymbol{\varrho}$, is in a pure state. If not then the quantum state is called a *mixed state*.

Now that any quantum system can be represented by a positive operator $\boldsymbol{\varrho}$ the set of possible transformations which describe the dynamics of the system (time evolution/measurement) has to be extended. This dynamics is characterized by so called *quantum operations* $\boldsymbol{\mathcal{E}}(\boldsymbol{\varrho})$ that map the initial state $\boldsymbol{\varrho}$ to the final state $\hat{\boldsymbol{\varrho}} = \boldsymbol{\mathcal{E}}(\boldsymbol{\varrho})$. Any quantum operation $\boldsymbol{\mathcal{E}}^A(\boldsymbol{\varrho}^A)$ on system S^A must be a completely positive map [NC00, Aud05]. This means that (1) every positive operator $\boldsymbol{\varrho}^A$ of the system S^A is mapped to another positive operator $\hat{\boldsymbol{\varrho}}^A = \boldsymbol{\mathcal{E}}^A(\boldsymbol{\varrho}^A)$ of system S^A and (2) if one introduces an extra system S^B then the quantum operation $\boldsymbol{\mathcal{E}}^A \otimes \mathbf{1}^B(\boldsymbol{\varrho}^{AB})$ maps any positive operator $\boldsymbol{\varrho}^{AB}$ of the combined system S^{AB} to another positive operator of the combined system. One also demands that $\boldsymbol{\mathcal{E}}^A$ is a linear operation. It is well known that for any quantum operation $\boldsymbol{\mathcal{E}}^A$ on system S^A there exists a unitary transformation $\boldsymbol{U}^{AB}$ on an extended system S^{AB} so that any state $\boldsymbol{\varrho}^A$ of system S^A transforms according to the quantum operation $\boldsymbol{\mathcal{E}}^A$ on system S^A alone [Aud05].

Up to now we only considered projective measurements that are non-destructive by definition (see Axiom 3 in Sec. 2.4). Nevertheless, experiments where particles are diffracted at a single slit and are finally detected on a detection screen are not captured by projective measurements. After the detection the particles are not accessible for any further measurements. To adequately describe such scenarios one uses the concept of POVM (**p**ositive **o**perator-**v**alued **m**easure) measurements [NC00, Aud05]. Knowing the set $\{m_i\}$, with $m_i \in \mathbb{R}$, of possible measurement results and the corresponding probabilities p_i it is possible to assign a linear operator $\boldsymbol{E}_{m_i}$ to each measurement result m_i. Due to $p_i = tr\{\boldsymbol{\varrho}\boldsymbol{E}_{m_i}\}$ and $0 \leq p_i \leq 1$ the operators $\boldsymbol{E}_{m_i}$ must be positive. Because of $\sum p_i = 1$ and the linearity of the trace operation one gets $\sum_i \boldsymbol{E}_{m_i} = \mathbf{1}$. It is well known that for any POVM measurement on system S^A there exist unitary transformations $\boldsymbol{U}^{AB}$ on an extended system S^{AB} and projective measurements on system S^B so that system S^A is measured according to the POVM formalism [Aud05].

The scheme for realizing a POVM measurement described in the last sentence nicely reflects the physical events that occur when a particle in the single slit experiment described above hits the detection screen. Not the particle is measured but the photon that is emitted when the particle hits the screen. So it becomes apparent that any description of such a measurement process has to consider the emitted photon as well. This is exactly what the POVM formalism provides.

The most general dynamics of a quantum system S^A can be represented by unitary transformations $\boldsymbol{U}^{AB}$ on an extended system S^{AB} and projective measurements on system S^B. Hence, it is sufficient to only simulate unitary transformations and projective measurements in order to adequately simulate quantum computers on classical computers.

2.8 Liquid State NMR Quantum Computers

Up to now NMR-QCs provide the only possibility to realize QAs on several qubits, e.g. Shor's algorithm for seven qubits was implemented on an NMR-QC [VSB+01]. As the parity algorithm found by our GP system was implemented on a two-qubit NMR-QC we will present the theory of NMR-QCs by means of a two-qubit system.

In Fig. 2.2 a sample with a macroscopic number of ^{13}C-labeled chloroform molecules is placed into a strong uniform magnetic field $\underline{B}_0$, conventionally taken to define the $\boldsymbol{e}_z$-axis:

$$\underline{B}_0 = B_0 \boldsymbol{e}_z.$$

Perpendicular to this constant magnetic field a coil is oriented along the $\boldsymbol{e}_x$-axis. This coil allows to create a small oscillating magnetic field $\underline{B}_1$ that can rapidly be switched on and off [Sli90]:

$$\underline{B}_1(t) = \frac{B_1}{2}\left[(\cos(\omega_{rf}t)\boldsymbol{e}_x - \sin(\omega_{rf}t)\boldsymbol{e}_y) + \underbrace{(\cos(\omega_{rf}t)\boldsymbol{e}_x + \sin(\omega_{rf}t)\boldsymbol{e}_y)}_{\text{negligible}}\right]. \quad (2.15)$$

The Hamiltonian that describes the two spin-1/2 nuclei ^{13}C and ^{1}H of a single chloroform molecule follows from Pauli's equation Eq. (2.10):

$$\boldsymbol{H}(t) = \boldsymbol{H}_0 + \boldsymbol{H}_{rf}(t). \quad (2.16)$$

Here $\boldsymbol{H}_0$ denotes the time independent Hamiltonian of the system with the radiofrequency (rf) field switched off:

$$\boldsymbol{H}_0 = -\omega_0^{(H)} \boldsymbol{I}_z^{(H)} - \omega_0^{(C)} \boldsymbol{I}_z^{(C)} + 2\pi J\, \boldsymbol{I}_z^{(H)} \boldsymbol{I}_z^{(C)}. \quad (2.17)$$

Above we introduced $\boldsymbol{I}_\alpha^{(H)} = 1/2(\boldsymbol{\sigma}_\alpha \otimes \mathbf{1})$ with $\alpha \in \{x, y, z\}$ to denote the spin operator of the hydrogen nucleus and $\boldsymbol{I}_\alpha^{(C)} = 1/2(\mathbf{1} \otimes \boldsymbol{\sigma}_\alpha)$ to denote the spin operator of the carbon nucleus. Therefore, $2\pi J\, \boldsymbol{I}_z^{(H)} \boldsymbol{I}_z^{(C)}$ is equal to $(\pi/2) J\, \boldsymbol{\sigma}_z \otimes \boldsymbol{\sigma}_z$.

The term $\boldsymbol{H}_{rf}$ denotes the time dependent part of the Hamiltonian due to the rf-field:

$$\begin{aligned} \boldsymbol{H}_{rf}(t) \quad = \quad & -\omega_1^{(H)} \cdot \left(\cos(\omega_{rf}t)\boldsymbol{I}_x^{(H)} - \sin(\omega_{rf}t)\boldsymbol{I}_y^{(H)}\right) \\ & -\omega_1^{(C)} \cdot \left(\cos(\omega_{rf}t)\boldsymbol{I}_x^{(C)} - \sin(\omega_{rf}t)\boldsymbol{I}_y^{(C)}\right). \end{aligned} \quad (2.18)$$

We introduced the two constants ω_0, ω_1 that are defined by $\omega_0 = \gamma \cdot B_0$ and $\omega_1 = \gamma \cdot B_1/2$, where γ is the gyromagnetic ratio of the considered nuclei. The

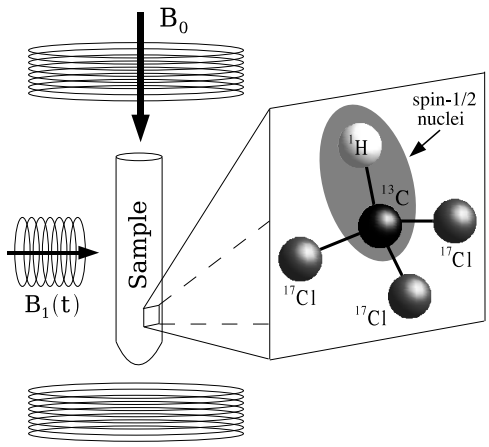

Figure 2.2: Schematic diagram of an NMR-QC.

term $2\pi J\, \boldsymbol{I}_z^{(H)} \boldsymbol{I}_z^{(C)}$ follows from the scalar coupling between the spin of the hydrogen nucleus and that of the carbon nucleus [Hor95]. For chloroform one gets J=215 Hz.

The time independent part $\boldsymbol{H}_0$ of the Hamiltonian in Eq. (2.16) is diagonal in the computational basis which is defined by the eigenstates of the $\boldsymbol{I}_z^{(H)}$ and $\boldsymbol{I}_z^{(C)}$ operators:

$$\boldsymbol{H}_0 = \begin{pmatrix} E_0 & 0 & 0 & 0 \\ 0 & E_1 & 0 & 0 \\ 0 & 0 & E_2 & 0 \\ 0 & 0 & 0 & E_3 \end{pmatrix},$$

with

$$\begin{aligned} E_0 &= (-\omega_0^{(H)} - \omega_0^{(C)} + \pi J)/2, \quad & E_1 &= (-\omega_0^{(H)} + \omega_0^{(C)} - \pi J)/2, \\ E_2 &= (\omega_0^{(H)} - \omega_0^{(C)} - \pi J)/2, & E_3 &= (\omega_0^{(H)} + \omega_0^{(C)} + \pi J)/2. \end{aligned} \tag{2.19}$$

2.8.1 Controllable Unitary Operations

To get rid of the explicit time dependence in the Schrödinger equation:

$$\imath\hbar\partial_t|\psi(t)\rangle = \boldsymbol{H}(t)|\psi(t)\rangle,$$

we use the following formula [Sor89]:

$$e^{-\imath\varphi\boldsymbol{B}}\boldsymbol{A}e^{\imath\varphi\boldsymbol{B}} = \boldsymbol{A}\cos(\varphi) - \imath\,[\boldsymbol{B},\boldsymbol{A}]\sin(\varphi),$$

which holds for $[\boldsymbol{B},[\boldsymbol{B},\boldsymbol{A}]] = \boldsymbol{A}$, where $[\boldsymbol{A},\boldsymbol{B}] = \boldsymbol{A}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{A}$. The operators $\boldsymbol{A}$ and $\boldsymbol{B}$ denote spin operators $\boldsymbol{I}_\alpha^{(H)}$, $\boldsymbol{I}_\beta^{(C)}$ and products thereof, with $\alpha,\beta \in \{x,y,z\}$. Due to this formula the Hamiltonian in Eq. (2.16) can be written in the form:

$$\begin{aligned} \boldsymbol{H} = \;& - \;\omega_0^{(H)}\boldsymbol{I}_z^{(H)} - \omega_0^{(C)}\boldsymbol{I}_z^{(C)} + 2\pi J\,\boldsymbol{I}_z^{(H)}\boldsymbol{I}_z^{(C)} \\ & - \;\omega_1^{(H)}\left(e^{\imath\omega_{rf}t\boldsymbol{I}_z^{(H)}}\boldsymbol{I}_x^{(H)}e^{-\imath\omega_{rf}t\boldsymbol{I}_z^{(H)}}\right) - \omega_1^{(C)}\left(e^{\imath\omega_{rf}t\boldsymbol{I}_z^{(C)}}\boldsymbol{I}_x^{(C)}e^{-\imath\omega_{rf}t\boldsymbol{I}_z^{(C)}}\right). \end{aligned}$$

The explicit time dependence of the Hamiltonian vanishes if one uses the following substitution to solve the Schrödinger equation:

$$|\psi(t)\rangle = e^{\imath\omega_{rf}t\boldsymbol{I}_z^{(H)}} e^{\imath\omega_{rf}t\boldsymbol{I}_z^{(C)}} |\varphi(t)\rangle.$$

This transformation is equivalent to a change of the frame of reference from the static laboratory frame to a rotating frame that rotates around the $\boldsymbol{e}_z$-axis with frequency ω_{rf} [Sli90]. From now on we integrate Planck's constant into the Hamiltonian $\boldsymbol{H}$. One thus gets:

$$\imath\partial_t|\varphi(t)\rangle = \boldsymbol{H}'|\varphi(t)\rangle,$$

with:

$$\boldsymbol{H}' = \left(-\omega_0^{(H)} + \omega_{rf}\right)\boldsymbol{I}_z^{(H)} + \left(-\omega_0^{(C)} + \omega_{rf}\right)\boldsymbol{I}_z^{(C)} + 2\pi J\,\boldsymbol{I}_z^{(H)}\boldsymbol{I}_z^{(C)} - \omega_1^{(H)}\boldsymbol{I}_x^{(H)} - \omega_1^{(C)}\boldsymbol{I}_x^{(C)}.$$

Due to $|-\omega_0^{(H)} + \omega_0^{(C)}| \gg \omega_1^{(H)}$ the term proportional to $\omega_1^{(H)}$ can be neglected when ω_{rf} equals $\omega_0^{(C)}$. Because of $\omega_1^{(C)} \gg J$ one also can neglect the spin-spin coupling. For $\omega_{rf} = \omega_0^{(C)}$ one gets:

$$\imath\partial_t|\varphi(t)\rangle = \left((-\omega_0^{(H)} + \omega_0^{(C)})\boldsymbol{I}_z^{(H)} - \omega_1^{(C)}\boldsymbol{I}_x^{(C)}\right)|\varphi(t)\rangle.$$

Now the system's time evolution is described by:

$$\boldsymbol{U}(t) = e^{-\imath(\omega_0^{(H)} - \omega_0^{(C)})\boldsymbol{I}_z^{(H)}t} \cdot e^{\imath\omega_1^{(C)}\boldsymbol{I}_x^{(C)}t}.$$

The spin of the carbon nucleus evolves according to the unitary matrix $\boldsymbol{U}'(t)$:

$$\boldsymbol{U}'(t) = e^{\imath\omega_1^{(C)}\boldsymbol{I}_x^{(C)}t} = \boldsymbol{R}_x^{(C)}(-\omega_1^{(C)}t) = e^{\imath(\omega_1^{(C)}/2)t\,\mathbf{1}\otimes\boldsymbol{\sigma}_x}. \tag{2.20}$$

If one shifts the phase of the rf-field in Eq. (2.15) by an amount of $+90°$ then the spin of the carbon nucleus evolves according to the unitary matrix $\boldsymbol{U}'(t)$:

$$\boldsymbol{U}'(t) = e^{\imath\omega_1^{(C)}\boldsymbol{I}_y^{(C)}t} = \boldsymbol{R}_y^{(C)}(-\omega_1^{(C)}t) = e^{\imath(\omega_1^{(C)}/2)t\,\mathbf{1}\otimes\boldsymbol{\sigma}_y}. \tag{2.21}$$

Due the results presented above and Eq. (3.11) of Sec. 3.3 any unitary operation on a single qubit can be realized by appropriately chosen phase shifts and frequencies of the rf-field. According to Sec. 3.3 any unitary operation on n qubits can be decomposed into unitary operations on single qubits and a unitary operation on two qubits called CNOT:

$$\boldsymbol{U}_{CNOT}^{(HC)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Here we used $\boldsymbol{U}_{CNOT}^{(HC)}$ to denote that the hydrogen's spin provides the control and the carbon's spin the target qubit: $\boldsymbol{U}_{CNOT}^{(HC)}|a\rangle|b\rangle = |a\rangle|b \oplus a\rangle$ with $a, b \in \{0, 1\}$ and $\oplus$ the XOR operation. The leftmost qubit is the spin of the hydrogen nuclei. Thus, the spin state of the carbon nuclei is flipped only when the hydrogen's spin is in state $|1\rangle$.

This gate can be realized up to a global phase shift by rf-pulses and periods of free evolution where the system evolves due to the time independent Hamiltonian $\boldsymbol{H}_0$:

$$\imath\partial_t|\psi(t)\rangle = \boldsymbol{H}_0|\psi(t)\rangle.$$

Substituting $|\psi(t)\rangle$ by $|\varphi(t)\rangle$ according to:

$$|\psi(t)\rangle = e^{\imath\omega_0^{(H)}t\boldsymbol{I}_z^{(H)}}e^{\imath\omega_0^{(C)}t\boldsymbol{I}_z^{(C)}}|\varphi(t)\rangle,$$

which is equivalent to change the frame of reference to the doubly rotating frame with frequencies $\omega_0^{(H)}$ and $\omega_0^{(C)}$, one gets:

$$\imath\partial_t|\varphi(t)\rangle = 2\pi J\,\boldsymbol{I}_z^{(H)}\boldsymbol{I}_z^{(C)}|\varphi(t)\rangle.$$

In the doubly rotating frame the time evolution caused by the time independent Hamiltonian $\boldsymbol{H}_0$ is described by the unitary matrix $\boldsymbol{U}(t)$:

$$\boldsymbol{U}(t) = e^{-\imath 2\pi J\boldsymbol{I}_z^{(H)}\boldsymbol{I}_z^{(C)}t} = \boldsymbol{R}_{zz}^{(HC)}(2\pi Jt) = e^{-\imath\frac{\pi}{2}Jt\,\boldsymbol{\sigma}_z\otimes\boldsymbol{\sigma}_z}. \tag{2.22}$$

Using $\boldsymbol{R}_x^{(i)}(\phi)$, $\boldsymbol{R}_y^{(i)}(\phi)$ and $\boldsymbol{R}_{zz}^{(ij)}(\phi)$, defined in Eq. (2.20) - Eq. (2.22), with $i,j \in \{H,C\}$ one gets, up to a global phase shift:

$$\boldsymbol{U}_{CNOT}^{(HC)} \sim \boldsymbol{R}_y^{(C)}(\frac{\pi}{2})\cdot\boldsymbol{R}_{-x}^{(C)}(\frac{\pi}{2})\cdot\boldsymbol{R}_{-y}^{(C)}(\frac{\pi}{2})\cdot\boldsymbol{R}_x^{(H)}(\frac{\pi}{2})\cdot\boldsymbol{R}_{-y}^{(H)}(\frac{\pi}{2})\cdot\boldsymbol{R}_{zz}^{(HC)}(\pi)\cdot\boldsymbol{R}_y^{(H)}(\frac{\pi}{2}). \tag{2.23}$$

Any one-qubit and CNOT operation can be realized on the NMR-QC. It follows that any unitary two-qubit operation can be realized (see Sec. 3.3). This result can also be generalized to an arbitrary number of qubits. Nevertheless, for an n-qubit system it is necessary to get rid of unwanted spin-spin interactions which can be done using refocusing pulses [JK99]. In the calculations above we often switched the frame of reference which, in the experiment, corresponds to a phase shift in the measured signal.

In the NMR literature the rf-pulses are usually denoted by the following symbols:

$$\begin{aligned}
&(\theta)_{\pm x}^{(C)} = \boldsymbol{R}_{\pm x}^{(C)}(\theta), \quad (\theta)_{\pm y}^{(C)} = \boldsymbol{R}_{\pm y}^{(C)}(\theta), \qquad (\theta)_{\pm x}^{(H)} = \boldsymbol{R}_{\pm x}^{(H)}(\theta),\\
&(\theta)_{\pm y}^{(H)} = \boldsymbol{R}_{\pm y}^{(H)}(\theta), \quad (\theta)^{(HC)} = \boldsymbol{R}_{zz}^{(HC)}(\theta), \quad \theta\in\mathbb{R}_+
\end{aligned} \tag{2.24}$$

Pulse sequences are read from the left to the right. The pulse sequence to implement the $\boldsymbol{U}_{CNOT}^{(HC)}$ gate in Eq. (2.23) reads:

$$\boldsymbol{U}_{CNOT}^{(HC)} \equiv \left(\tfrac{\pi}{2}\right)_y^{(H)} - (\pi)^{(HC)} - \left(\tfrac{\pi}{2}\right)_{-y}^{(H)} - \left(\tfrac{\pi}{2}\right)_x^{(H)} - \left(\tfrac{\pi}{2}\right)_{-y}^{(C)} - \left(\tfrac{\pi}{2}\right)_{-x}^{(C)} - \left(\tfrac{\pi}{2}\right)_y^{(C)}.$$

2.8.2 The Thermal Density Matrix

Dipole-dipole interactions between intra- and intermolecular spins are averaged out due to rapid tumbling of the molecules in the macroscopic sample at room temperature [Lev01]. Therefore, these interactions can be neglected.

The probability p_i of the two spin-1/2 system to be in the energy eigenstate $|\underline{i}\rangle$ with $\boldsymbol{H}_0|\underline{i}\rangle = E_i|\underline{i}\rangle$ is given by Boltzmann's distribution:

$$p_i = \frac{e^{-\beta E_i}}{\sum_j p_j}.$$

Here $\beta = 1/kT$ with k the Boltzmann constant and T the temperature.

The whole ensemble of chloroform molecules subjected to a constant magnetic field is described by the density matrix $\boldsymbol{\varrho}_{therm}$:

$$\boldsymbol{\varrho}_{therm} = \sum_i p_i |\underline{i}\rangle\langle\underline{i}| = \sum_i \frac{e^{-\beta E_i}}{\sum_j e^{-\beta E_j}} |\underline{i}\rangle\langle\underline{i}| = \frac{e^{-\beta \boldsymbol{H}_0}}{tr\{e^{-\beta \boldsymbol{H}_0}\}}.$$

In the high temperature approximation one obtains:

$$\boldsymbol{\varrho}_{therm} = \frac{e^{-\beta \boldsymbol{H}_0}}{tr\{e^{-\beta \boldsymbol{H}_0}\}} \approx \frac{1}{4}\left(\mathbf{1} - \beta \boldsymbol{H}_0\right).$$

Due to $\omega_0^{(H)}, \omega_0^{(C)} \gg 2\pi J$ one can neglect the spin-spin coupling:

$$\boldsymbol{\varrho}_{therm} \approx \frac{1}{4}\left(\mathbf{1} + \beta\omega_0^{(H)} \boldsymbol{I}_z^{(H)} + \beta\omega_0^{(C)} \boldsymbol{I}_z^{(C)}\right). \tag{2.25}$$

2.8.3 The Measurement

With the rf-field switched off the time dependence $|\psi(t)\rangle$ of a single two spin-1/2 system is described by:

$$|\psi(t)\rangle = \sum_l c_l(t)|\underline{l}\rangle = e^{-\imath \boldsymbol{H}_0 t} \sum_l c_l(0)|\underline{l}\rangle = \sum_l e^{-\imath E_l t} c_l(0)|\underline{l}\rangle, \quad \text{with } l \in \{0,1,2,3\}.$$

The energies E_l are specified in Eq. (2.19). The time evolution of the density matrix $\boldsymbol{\varrho}(t)$ of a single system in the pure state $|\psi(t)\rangle$ can be calculated via:

$$\boldsymbol{\varrho}(t) = |\psi(t)\rangle\langle\psi(t)| = \sum_{l,m=0} c_l(0)c_m^*(0)e^{-\imath\omega_{lm}t}|\underline{l}\rangle\langle\underline{m}| \quad \text{with } \omega_{lm} = E_l - E_m.$$

The time evolution of the density matrix element $(\boldsymbol{\varrho}(t))_{lm}$ is described by:

$$(\boldsymbol{\varrho}(t))_{lm} = e^{-\imath\omega_{lm}t}(\boldsymbol{\varrho}(0))_{lm}.$$

Due to the definition of the general density matrix in Eq. (2.14) this equation also holds for mixed states. For pure states one gets:

$$(\boldsymbol{\varrho}(t))_{lm} = c_l(t)c_m^*(t).$$

The voltage $s_{FID}(t)$ generated in the coil oriented along $\boldsymbol{e}_x$ is proportional to the time derivative of the NMR sample's magnetization along the $\boldsymbol{e}_x$-axis:

$$s_{FID}(t) \sim \partial_t\langle \boldsymbol{I}_x^{(H)} + \boldsymbol{I}_x^{(C)}\rangle(t) = tr\left\{(\partial_t \boldsymbol{\varrho}(t))\left(\boldsymbol{I}_x^{(H)} + \boldsymbol{I}_x^{(C)}\right)\right\}.$$

As $\boldsymbol{\varrho}$ is a Hermitian operator $(\boldsymbol{\varrho})_{lm}$ is equal to $(\boldsymbol{\varrho})^*_{ml}$. From the definition of ω_{lm} above it follows that ω_{lm} is equal to $-\omega_{ml}$. One gets:

$$\begin{aligned} s_{FID}(t) \quad \sim \quad & \omega_{01} \cdot \Im\{(\boldsymbol{\varrho}(0))_{01} \cdot e^{-\imath\omega_{01}t}\} - \omega_{02} \cdot \Im\{(\boldsymbol{\varrho}(0))_{02} \cdot e^{-\imath\omega_{02}t}\} \\ & -\omega_{13} \cdot \Im\{(\boldsymbol{\varrho}(0))_{13} \cdot e^{-\imath\omega_{13}t}\} - \omega_{23} \cdot \Im\{(\boldsymbol{\varrho}(0))_{23} \cdot e^{-\imath\omega_{23}t}\}, \end{aligned}$$

with

$$\begin{aligned} \omega_{01} &= -\omega_0^{(C)} + \pi J, \quad \omega_{02} = -\omega_0^{(H)} + \pi J \\ \omega_{13} &= -\omega_0^{(H)} - \pi J, \quad \omega_{23} = -\omega_0^{(C)} - \pi J. \end{aligned}$$

We used $\Im(z)$ to denote the imaginary part of a complex number z. Relaxation effects are not considered in this calculation hence one multiplies each addend in s_{FID} by a factor $\exp(-\lambda t)$, with $\lambda \in \mathbb{R}$. The thermal density matrix $\boldsymbol{\varrho}_{therm}$ in Eq. (2.25) has no outer-diagonal elements therefore one has to apply a $\boldsymbol{R}_y(\pi/2)$ pulse to both spins in order to measure any signal:

$$\boldsymbol{U}\boldsymbol{\varrho}_{therm}\boldsymbol{U}^+ \approx \frac{1}{4}\left(\mathbf{1} + \beta\omega_0^{(H)}\boldsymbol{I}_x^{(H)} + \beta\omega_0^{(C)}\boldsymbol{I}_x^{(C)}\right),$$

with $\boldsymbol{U}$ defined by:

$$\boldsymbol{U} = \boldsymbol{R}_y^{(H)}(\pi/2)\boldsymbol{R}_y^{(C)}(\pi/2).$$

This returns $(\boldsymbol{\varrho}(0))_{01} = (\boldsymbol{\varrho}(0))_{23} = \omega_0^{(C)}/2$ and $(\boldsymbol{\varrho}(0))_{02} = (\boldsymbol{\varrho}(0))_{13} = \omega_0^{(H)}/2$.

In order to visualize the individual components of the measured signal $s_{FID}(t)$ one calculates the Fourier transform $S(\Omega)$ of $s_{FID}(t)$:

$$S(\Omega) = \int_0^\infty s_{FID}(t)e^{-\imath\Omega t}\,dt.$$

The spectrum $S(\Omega)$ of the signal $s_{FID}(t)$:

$$s_{FID}(t) = \sum_l a_l \cdot s_l(t) \quad \text{with} \quad s_l(t) = e^{-(\imath\omega_l + \lambda_l)t} \quad \text{and} \quad a_l, \omega_l, \lambda_l \in \mathbb{R},$$

is of the form [Lev01]:

$$S(\Omega) = \sum a_l \left(\underbrace{\frac{\lambda_l}{\lambda_l^2 + (\Omega - \omega_l)^2}}_{\text{absorption Lorentzian}} - \imath \cdot \underbrace{\frac{(\Omega - \omega_l)}{\lambda_l^2 + (\Omega - \omega_l)^2}}_{\text{dispersion Lorentzian}} \right).$$

Applying $\boldsymbol{R}_y(\pi/2)$ pulses to both spins of the thermally equilibrated system in Fig. 2.2 results in the spectrum shown in Fig. 2.3.

2.8.4 Pseudo-Pure States and Temporal Averaging

Usually, a quantum computation starts with a pure state of the form $|\psi_{init}\rangle = |\underline{0}\rangle$. In Sec. 2.8.2 we showed that the initial state of an NMR-QC is described by the thermal

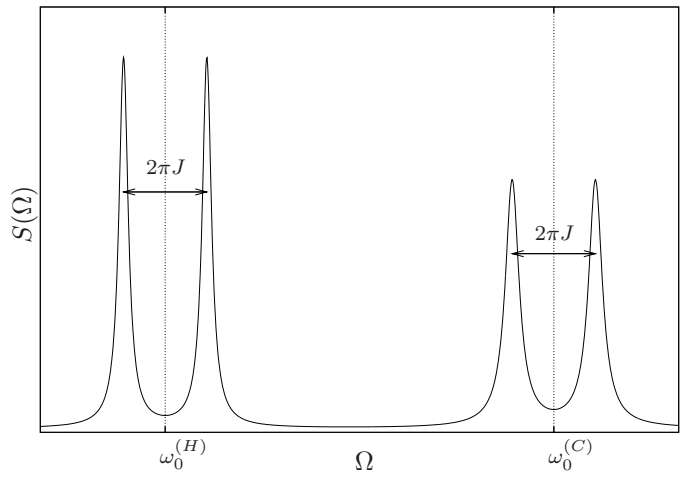

Figure 2.3: Schematic diagram of the real part of the NMR spectrum as it would be obtained after applying a readout pulse ($\boldsymbol{R}_y(\pi/2)$-pulse) to the equilibrated NMR sample of Fig. 2.2.

density matrix $\boldsymbol{\varrho}_{therm}$. Nevertheless, it is possible to perform a quantum computation as the thermal density matrix can be transformed into a state of the form:

$$\boldsymbol{\varrho} = (1-\epsilon)\frac{\mathbf{1}}{N} + \epsilon|\underline{0}\rangle\langle\underline{0}|,$$

also called pseudo-pure state, where $N = 2^n$ denotes the dimension of the Hilbert space of an n-qubit system. A computation represented by the unitary transformation $\boldsymbol{U}$ maps this density matrix to:

$$\boldsymbol{U}\boldsymbol{\varrho}\boldsymbol{U}^+ = (1-\epsilon)\frac{\mathbf{1}}{N} + \epsilon\boldsymbol{U}|\underline{0}\rangle\langle\underline{0}|\boldsymbol{U}^+.$$

A readout pulse $\boldsymbol{R}_y(\pi/2)$ leaves the term proportional to $\mathbf{1}$ unchanged. This term does not contribute to the magnetization along the $\boldsymbol{e}_x$-axis. Therefore, the whole signal is only caused by the last term. Unfortunately, the factor ϵ decreases exponentially with the number n of qubits and thus renders the usage of pseudo-pure states ineffective for large n [WGC97].

In our experiment we used a method called temporal averaging to create a pseudo-pure state [KCL98]. Here we illustrate this approach for the two-qubit system of Fig. 2.2. As shown in Sec. 2.8.2 the thermal state of our system is:

$$\boldsymbol{\varrho}_{\text{therm}} \approx 2^{-2}[\mathbf{1} - \boldsymbol{H}_0] \approx 2^{-2}[\mathbf{1} + \omega_0^{(A)}\boldsymbol{I}_z^{(A)} + \omega_0^{(X)}\boldsymbol{I}_z^{(X)}] = \begin{pmatrix} n_{00} & & & \\ & n_{01} & & \\ & & n_{10} & \\ & & & n_{11} \end{pmatrix},$$

with $n_{00}, n_{01}, n_{10}, n_{11} \in \mathbb{R}$. Since this matrix only consists of diagonal elements we will abbreviate it by $[n_{00},n_{01},n_{10},n_{11}]$. A pseudo-pure state of the form:

$$\boldsymbol{\varrho} = (n_{01}+n_{10}+n_{11})[1,1,1,1] + (3n_{00} - (n_{01}+n_{10}+n_{11}))\,[1,0,0,0],$$

is equivalent to the sum of three diagonal density matrices:

$$\begin{aligned}&[n_{00}, n_{01}, n_{10}, n_{11}] + [n_{00}, n_{11}, n_{01}, n_{10}] + [n_{00}, n_{10}, n_{11}, n_{01}] \\ &= \quad (n_{01} + n_{10} + n_{11})[1,1,1,1] + (3n_{00} - (n_{01} + n_{10} + n_{11}))\,[1,0,0,0]. \quad (2.26)\end{aligned}$$

The first addend on the left is the equilibrium density matrix $\boldsymbol{\varrho}_{therm}$ and the last two addends on the left can be created by applying $\boldsymbol{CNOT}$ operations to $\boldsymbol{\varrho}_{therm}$ in the following manner:

$$\begin{aligned}[n_{00}, n_{11}, n_{01}, n_{10}] &= \boldsymbol{U}_{CNOT}^{(CH)} \cdot \boldsymbol{U}_{CNOT}^{(HC)} \cdot [n_{00}, n_{01}, n_{10}, n_{11}] \cdot \boldsymbol{U}_{CNOT}^{(HC)\ +} \cdot \boldsymbol{U}_{CNOT}^{(CH)\ +}, \\ [n_{00}, n_{10}, n_{11}, n_{01}] &= \boldsymbol{U}_{CNOT}^{(HC)} \cdot \boldsymbol{U}_{CNOT}^{(CH)} \cdot [n_{00}, n_{01}, n_{10}, n_{11}] \cdot \boldsymbol{U}_{CNOT}^{(CH)\ +} \cdot \boldsymbol{U}_{CNOT}^{(HC)\ +}. \end{aligned} \quad (2.27)$$

An NMR-experiment is mathematically described by linear operations. Hence, an initial state like the one on the right side of Eq. (2.26) is equivalent to the sum of the spectra of three different experiments, each of which is performed on one of the three initial states similar to those on the left side of Eq. (2.26).

3 Classical and Quantum Computation

One major problem one is faced with in the investigation of QAs is the fact that almost all QAs of interest like the Deutsch-Jozsa (DJ) algorithm [DJ92], Simon's algorithm [Sim94], Grover's algorithm [Gro96] or order-finding[1] are naturally stated using the so called blackbox model of computation [BBC+01]. The application range of this model is very restricted in comparison to the usual model of computation, namely Turing's machine. Despite some proofs of an exponential speed-up of QAs for blackbox problems like Simon's problem (see Sec. 3.4.2) or order finding [Cle99] it remains unclear if quantum computation is superior to classical computation (see Sec. 3.1.4).

The aim of this chapter is to provide the main concepts of computability and computational complexity in order to put the blackbox model of computation in perspective. This model will be used to discuss the QAs evolved with the help of GP in Chap. 5.

We will start with a general introduction to computation in Sec. 3.1. Thereafter we will present deterministic computation in Sec. 3.1.1, probabilistic computation in Sec. 3.1.2, and quantum computation in Sec. 3.1.3. Unless stated otherwise the definitions and concepts are taken from the textbooks of Gruska [Gru00] and Hirvensalo [Hir01].

The concept of computational complexity will be presented in Sec. 3.1.4. The definitions of the classical complexity classes are taken from [Vöc02]. In Sec. 3.1.5 we will discuss relativized complexity classes. They can be used to demonstrate that quantum computation indeed provides some benefits over classical computation. This section will also introduce the concept of oracles, or analogous blackboxes, that gives rise to the blackbox model of computation that will be presented in Sec. 3.2.

The latter model of computation uses decision trees to keep track of the different results returned by oracle queries. Quantum decision trees can be defined by a sequence of oracle calls alternating with unitary matrices [BdW02]. These matrices can be decomposed into one- and two-qubit operations. Therefore, we will introduce the concept of quantum circuits in Sec. 3.3. Finally, Sec. 3.4 will present the DJ algorithm and Simon's algorithm.

3.1 Turing Machines and Computational Complexity Classes

The concept of computability is closely related to the Turing machine (TM). Turing's main concern in introducing this machine was on computable numbers [Tur36]. He also mentions that his computing machine can easily be extended to define computable

[1] Shor's algorithm [Sho94] reduces the factoring problem of integers to the problem of order-finding [NC00]. This reduction can be done efficiently on classical computers. Shor's algorithm thus exploits the ability of quantum computers to efficiently solve the order-finding problem.

functions, computable predicates, and so forth. Therefore, the TM can be used to formalize the notion *computable*.

His investigation was motivated by an outstanding problem in the foundations of mathematics called *decision problem* [Hod02] which was posed by David Hilbert. He worked on a formalistic approach on the foundations of mathematics also known as *Hilbert's program* [Zac03]. The decision problem is to show that there exists a definite method to decide with a finite number of elementary steps whether a mathematical proposition is provable or not.

The main difficulty in solving this problem was to find a general definition of notions usually used in Hilbert's program like "definite method" or "procedure". To solve this problem Turing tried to identify a minimal set of requirements and procedures needed by a person to perform a computation (see §9 in [Tur36]). He obtained the following list of essential conditions:

- A computation is done by writing certain symbols on paper which is divided into squares like a child's arithmetic book. Without loss of generality this two-dimensional paper can be represented by a one-dimensional one - i.e., a tape divided into squares.
- The number of symbols that may be printed is finite.
- The behavior of the person at any moment of the computation is determined by the symbols he currently observes and his corresponding "state of mind".
- The number of these states of mind is taken to be finite.
- The operations of the computing person can be split up into simple elementary operations so that no more than one symbol is altered per operation.
- The person can move from one square of the tape to a neighboring one at each elementary operation.

This informal description provides the main ideas necessary to formalize the notion of computability. In the next section we demonstrate how this can be done. There we will define the *deterministic TM* (DTM).

Devices that implement Turing's list of essential conditions are said to operate by *finite means* [Tur36]. This indicates that only a finite subsystem of the device is changed at every timestep. Moreover, the actions of the device are specified by a finite number of rules that depend on the state of a finite subsystem.

3.1.1 The Deterministic Turing Machine

A DTM consists of a sufficiently long tape[2], a read-write head and a finite state machine (see Fig. 3.1). The tape is divided into an infinite number of cells. Each of these cells can either be blank or contain a symbol σ from a finite set Σ also called alphabet. The read-write head can access only one cell per time-step. It is controlled by a finite state machine whose actual state s is an element of a finite set S of possible states. Dependent on the symbol σ and the actual state s, the TM enters the new

[2] Usually the tape is considered to be of infinite length. Nevertheless, only a finite number of tape cells is allowed to contain symbols relevant to the computation.

state $s' \in S$. Then it replaces the symbol σ by $\sigma' \in \Sigma$ and moves the read-write head to a neighboring cell. The direction of this move is determined by the variable $d \in \{\mathrm{R}, \mathrm{L}\}$: For d=L the head moves to the left, for d=R it moves to the right. The time evolution of the DTM is described by the transition function δ_{DTM}:

$$\delta_{\mathrm{DTM}} : S \times \Sigma \to S \times \Sigma \times \{\mathrm{R}, \mathrm{L}\}.$$

In the process of computation the DTM goes through a sequence of *configurations*. Each configuration $c = (s_1, \underline{w}_1\sigma, \sigma_1\underline{w}_2)$ provides a global description of the machine. It is determined by the symbols $\sigma, \sigma_1 \in \Sigma$ written on the tape, the actual state $s_1 \in S$ of the finite state machine and the actual position of the read-write head. The read-write head currently reads the symbol σ_1. The symbols $\underline{w}_1$ and $\underline{w}_2$ denote *words* over the alphabet Σ. Usually one writes $\underline{w}_1, \underline{w}_2 \in \Sigma^*$ where Σ^* denotes the set of all possible words: If $\Sigma = \{0, 1\}$ then $\Sigma^* = \{0, 1, 01, 10, 11, 100, \ldots\}$.

The transition between configurations is completely determined by the transition function δ_{DTM}. If $\delta_{\mathrm{DTM}}(s_1, \sigma_1) = (s_2, \sigma_2, d)$ then the configuration $c = (s_1, \underline{w}_1\sigma, \sigma_1\underline{w}_2)$ can be transformed to:

$$\begin{aligned} c' &= (s_2, \underline{w}_1, \sigma\sigma_2\underline{w}_2) \quad \text{if } d = L. \\ c' &= (s_2, \underline{w}_1\sigma\sigma_2, \underline{w}_2) \quad \text{if } d = R. \end{aligned}$$

The transition function depends only on the actual state of the finite state machine and the current symbol read by the read-write head. Therefore the DTM operates by finite means as demanded in the end of the last section.

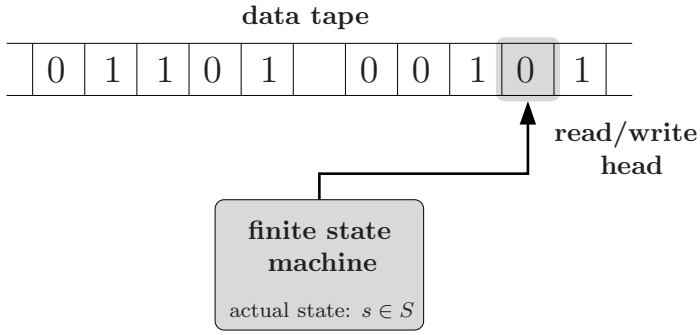

Figure 3.1: A DTM is described by the triplet M= (S, Σ, δ). S is a finite set of states, $\Sigma = \{0, 1\}$ is a finite set of symbols. The transition function $\delta_{\mathrm{DTM}} : S \times \Sigma \to S \times \Sigma \times \{\mathrm{R}, \mathrm{L}\}$ maps a configuration of the machine to the next one.

The DTM starts in the initial state $s_{\mathrm{init}} \in S$ with the read-write head at the first symbol of the input $\underline{w} \in \Sigma^*$. Starting in the initial configuration $c_{\mathrm{init}} = (s_{\mathrm{init}}, \epsilon, \underline{w})$ where ϵ denotes a blank region on the tape, the machine halts[3] in the configuration $c_{\mathrm{final}} = (s_{\mathrm{final}}, \underline{w}_1, \underline{w}_2)$. The result of the computation is the word $\underline{w}_1\underline{w}_2 \in \Sigma^*$. If the

[3] If one considers reversible TMs then the concept of a halting state has to be replaced by an "end of computation" flag. This flag ensures reversibility of the whole computation process, because a reversible non-trivial computation cannot have successive configurations that are equal.

DTM solves a decision problem then a single output symbol $\sigma_{\text{final}} \in \Sigma$ is sufficient: $\sigma_{\text{final}} = 0$ when the DTM rejects the input $\underline{w}$, $\sigma_{\text{final}} = 1$ when the DTM accepts the input $\underline{w}$.

The DTM seems to provide a sufficient model of computation. This conjecture is stated by the so-called Church-Turing thesis [NC00]:

> *The class of functions computable by a DTM corresponds to the class of functions which would naturally be regarded as being computable by an algorithm.*

One extension of the TM is due to a result of Solovay and Strassen who presented a new kind of algorithms for testing if a number is prime [SS77]. Their algorithm uses coin flips to help the search for counter-examples of primality. In order to cover such algorithms, Turing's original approach has been generalized to the *probabilistic TM* (PTM).

3.1.2 The Probabilistic Turing Machine

A PTM is a deterministic one augmented with the ability of an unbiased coin flip at each time step. Dependent on the outcome of this coin flip a deterministic transition function δ_{DTM} dictates the actual action of the machine. In Fig. 3.2 the coin flips are replaced by an additional tape that contains random bits. At each time step the machine reads a random bit from the additional tape and moves the corresponding read head to the neighboring right cell. One does not know the random bits in advance. Therefore, the behavior of the PTM is described by a transition function δ_{PTM} that assigns a probability $p \in [0,1]$ to the transition of a configuration $(s,\sigma) \in S \times \Sigma$ to a configuration $(s',\sigma') \in S' \times \Sigma'$, with $S = S'$ and $\Sigma = \Sigma'$:

$$\delta_{\text{PTM}} : S \times \Sigma \times S' \times \Sigma' \times \{\text{R},\text{L}\} \to p \;\text{ with }\; p \in \mathbb{R} \;\text{ and }\; p \in [0,1]. \tag{3.1}$$

The whole computation can be represented by a tree like the one in Fig. 3.4. Its nodes denote configurations and its edges assign transition probabilities. A single branch of this configuration tree corresponds to a single computation of the PTM. The total probability to end up in a particular final state is given by the product of the probabilities assigned to each edge on the path from the initial configuration to the final configuration. If several paths end up in the same final configuration then these probabilities have to be summed up.

The time evolution of the PTM can be described by stochastic matrices whose rows and columns are indexed by configurations. Each entry of the matrix assigns a transition probability from one configuration to another as demanded by Eq. (3.1). The stochastic matrix $\boldsymbol{P}$:

$$\boldsymbol{P} = \frac{1}{2}\begin{pmatrix} \mathbf{0} & \begin{matrix}1 & 1\\ 1 & 1\end{matrix} \\ \begin{matrix}1 & 1\\ 1 & 1\end{matrix} & \mathbf{0} \end{pmatrix} \qquad \text{with: } \mathbf{0} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

transforms an initial state $|c_1]$, represented by the vector $(1,0,0,0)^{\text{t}}$, to $\boldsymbol{P}\,|c_1] = 1/2\,|c_3] + 1/2\,|c_4]$, represented by $(0,0,1/2,1/2)^{\text{t}}$.

We use $|c_i]$ to denote the physical state that represents the configuration c_i. This notation is used to indicate that the corresponding physical system does not allow for

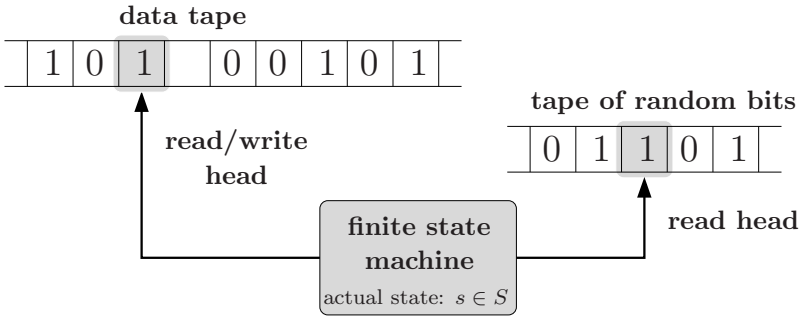

Figure 3.2: A PTM is a DTM whose transition function δ_{DTM} also depends on random bits read form an additional tape.

superpositions between different states. Therefore, the state $1/2\,|c_3] + 1/2\,|c_4]$ shows that the physical system is in the state $|c_3]$ or $|c_4]$ with a probability of $1/2$ each.

The configuration of a PTM at time t is calculated via:

$$|\psi(t)] = \boldsymbol{P}_t \boldsymbol{P}_{t-1} \cdots \boldsymbol{P}_1 |c_1] = p_1(t)\,|c_1] + p_2(t)\,|c_2] + \cdots + p_m(t)\,|c_m]. \tag{3.2}$$

Here m denotes the number of configurations and t the number of elementary computational steps. We used $\boldsymbol{P}_t, \ldots, \boldsymbol{P}_1$ to denote stochastic matrices. Because a TM has to work by finite means not all possible stochastic matrices represent valid TMs!

The next extension of TMs is due to Deutsch who developed a computation model based on quantum physics [Deu85]. He gave the first example of a problem that could be solved more adequately by this model than by any of the former approaches.[4]

The definition of the *quantum TM* (QTM) that will be presented in the next section is based on the work of Bernstein *et al.* who revised Deutsch's original approach [BV97].

3.1.3 The Quantum Turing Machine

A QTM is defined analogous to the DTM with the finite state machine, the tape, and the read-write head replaced by quantum systems. Due to the fact that the time evolution of a quantum system is described by unitary matrices $\boldsymbol{U}_i$ the transition function δ_{QTM} now looks like:

$$\delta_{\text{QTM}} : S \times \Sigma \times S' \times \Sigma' \times \{\text{R}, \text{L}\} \to \alpha \;\text{ with }\; \alpha \in \mathbb{C} \;\text{ and }\; |\alpha|^2 \in [0, 1].$$

Analogous to the time evolution of a PTM given by Eq. (3.2) the time evolution of a QTM is described by:

$$|\psi(t)\rangle = \boldsymbol{U}_t \boldsymbol{U}_{t-1} \ldots \boldsymbol{U}_1 |c_1\rangle = \alpha_1(t)|c_1\rangle + \alpha_2(t)|c_2\rangle + \cdots + \alpha_m(t)|c_m\rangle.$$

[4] The answer of Deutsch's algorithm to this problem is not obtained with certainty in a given time. Although his algorithm is sometimes faster than any classical algorithm it does not provide any speed-up on average.

Here we use Dirac's notation $|c_i\rangle$ to denote the quantum state that represents the ith configuration of the QTM. It can be decomposed in the following manner:

$$|c\rangle = |i\rangle|m\rangle|s\rangle; \quad i, m, s \in \mathbb{N}.$$

From now on we assume that the elementary computational states are represented by qubits, hence the configuration of the QTM is encoded binary. In what follows we use $\underline{i}$, $\underline{m}$ and $\underline{s}$ to denote the binary decomposition of the integers i, m and s. Therefore, the state $|\psi\rangle$ is denoted by $|\psi\rangle = |\underline{i}\rangle|\underline{m}\rangle|\underline{s}\rangle$. This indicates that the actual position of the read-write head accesses the i-th entry m_i of the memory (tape) $\underline{m} \in \{0,1\}^*$. The actual state of the finite state machine is given by the binary representation $\underline{s}$ of $s \in S$. The position i of the read-write head is increased by ± 1 at each time step. In addition to the read-write operations the QTM can apply any one-qubit operation to the memory entry at the actual position i of the read-write head.

The usage of physical systems governed by the rules of quantum physics allows to exploit new resources like superpositions and interference in the course of a computation. This is illustrated in Fig. 3.3.

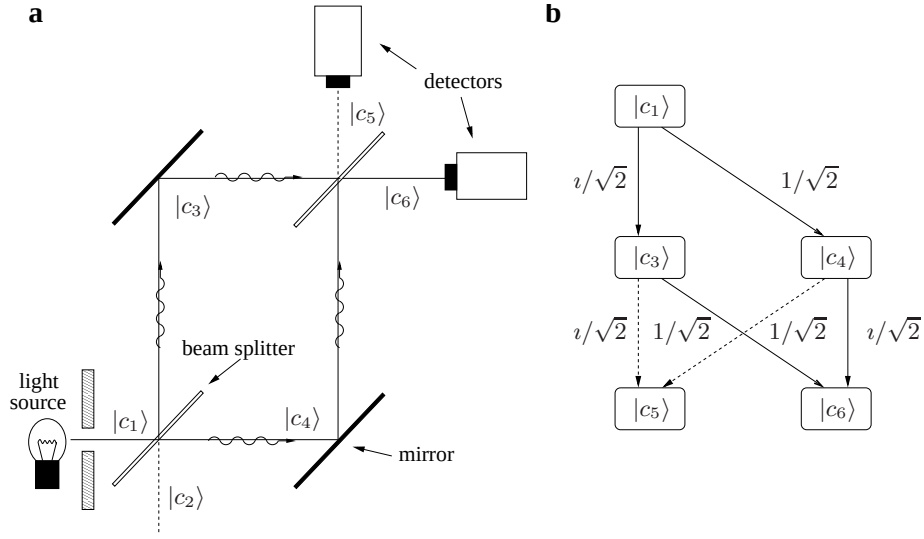

Figure 3.3: The experimental setup in part a illustrates the effects of superpositions and interference: Only the configuration $|c_6\rangle$ is measured. The corresponding configuration tree is shown in part b. The mirrors can be ignored because their effects in their respective arms balance out.

The first transition function of the computation performed by the experiment in Fig. 3.3a is represented by the unitary matrix $\boldsymbol{U}_1$ which describes the effect of the first beamsplitter. The mirrors can be ignored because their effects in their respective arms balance out. The final transition function (final beamsplitter) is represented by $\boldsymbol{U}_2$:

$$\boldsymbol{U}_1 = \begin{pmatrix} \mathbf{0} & \tilde{\boldsymbol{U}}_1 & \mathbf{0} \\ \tilde{\boldsymbol{U}}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}, \quad \boldsymbol{U}_2 = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\boldsymbol{U}}_1 \\ \mathbf{0} & \tilde{\boldsymbol{U}}_1 & \mathbf{0} \end{pmatrix}. \tag{3.3}$$

The unitary matrix $\tilde{\boldsymbol{U}}_1$, the identity operator $\mathbf{1}$ and the zero operator $\mathbf{0}$ are defined by:[5]

$$\tilde{\boldsymbol{U}}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \imath \\ \imath & 1 \end{pmatrix}, \quad \mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{0} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

The time evolution of the quantum system in Fig. 3.3a is described by $\boldsymbol{U}_2\boldsymbol{U}_1|c_1\rangle$ with $|c_1\rangle = (1,0,0,0,0,0)^{\mathrm{t}}$. The corresponding configuration tree is shown in Fig. 3.3b.

To calculate the total probability of ending up in a certain final quantum state one has to multiply the probability amplitudes assigned to each edge on the path from the root node to the final node. If several paths lead to the same quantum state then one has to sum up the corresponding probability amplitudes. The measurement probability of this quantum state is obtained by calculating the squared norm of the final probability amplitude.

Applying these rules to the configuration tree of the experiment demonstrates the effect of *interference*: Paths leading to the same final state may cancel out each other. In this example the probability to measure the configuration $|c_6\rangle = (0,0,0,0,0,1)^{\mathrm{t}}$ is equal to one.

If one considers the efficiency of a computation (see Sec. 3.1.4) then the usage of interference raises the hope to efficiently solve problems that were believed to be *computationally hard* in the classical approach on computability.

This can be illustrated by the concept of configuration trees. As will be stated in Sec. 3.1.4 a problem is efficiently solvable if at least more than one half of all possible paths of the configuration tree lead to an accepting configuration. If only a few paths lead to an accepting configuration then a problem is said to be computationally hard. This is due to the fact that the number of paths of a PTM grows exponentially with the number of time steps. A hard problem requires to test exponentially many paths on average in order to find an accepting one.

With the concept of QTMs it might be possible to boost the probability of finding an accepting path by causing destructive interference between the exponentially many rejecting ones.[6]

3.1.4 Computational Complexity

The main interest of Turing in introducing his computing machine was the investigation of computability. Therefore, he didn't account for the amount of time or memory needed by this machine. Nevertheless, considerations of these requirements became important in the investigation of "real-world" algorithms.

Hartmanis *et al.* introduced the idea of measuring the time and the space needed by the TM as a function of the length $|\underline{x}|$ of the input $\underline{x} \in \Sigma^*$ [HS65]:

- Time is measured by the number of elementary steps before the TM halts.
- Space is measured by the number of different character locations touched by the read-write head during a computation.

[5] A nice derivation of the form of the unitary matrix $\tilde{\boldsymbol{U}}_1$ can be found in [HGP02].

[6] This is only possible if the different computational paths have the same length and if each path does not leave around any garbage information. Paths with different length and different residual information would not interfere anymore. Further information about such requirements and the corresponding solutions can be found in Bernstein *et al.* [BV97].

J. Edmonds emphasized that the concept of *polynomial time* provides a good formalization of *efficient computation* because a wide range of tractable problems is computable in polynomial time [Edm65]. The concept of polynomial time indicates that the time t of the computation is upper bounded by a polynomial in the length $|\underline{x}|$ of the input $\underline{x}$: $t(|\underline{x}|) = poly(|\underline{x}|)$.[7]

These considerations led to the definition of the computational complexity class P of efficiently solvable problems.

Before we give the formal definition of this complexity class in Tab. 3.1 we have to introduce the concept of a *formal language* L: A language $\mathrm{L} \subset \Sigma^*$ is a subset of all words over the alphabet Σ that possess a certain property, e.g., the set of all binary strings that represent a prime number is a language $\mathrm{L} \subset \{0,1\}^*$.

Complexity Class **P**
A language L belongs to the complexity class P if there exists a DTM M so that for any input $\underline{x} \in \Sigma^*$ one of the following two conditions holds:
i) $\underline{x} \in \mathrm{L} \Rightarrow$ M accepts $\underline{x}$ in polynomial time $t = poly(\lvert\underline{x}\rvert)$.
ii) $\underline{x} \notin \mathrm{L} \Rightarrow$ M rejects $\underline{x}$ in polynomial time $t = poly(\lvert\underline{x}\rvert)$.

Table 3.1: Definition of the complexity class P.

For each language $\mathrm{L} \in \mathrm{P}$ there exists a DTM that decides in polynomial time if a word $\underline{x} \in \Sigma^*$ belongs to L or not.

The restriction on decision problems in the definition of complexity classes is not problematic as nearly every problem can be cast as a decision problem.

In the late 1970s it was realized that the complexity class P was too narrow to embrace all problems that are efficiently solvable. Especially the randomized algorithm to check primality of an integer proposed by Solovay *et al.* showed that probabilistic behavior has to be taken into account [SS77]. Therefore, one introduced additional complexity classes like BPP (**b**oundend-error **p**robability in **p**olynomial time). The definition of this complexity class can be found in Tab. 3.2.

Fig. 3.4 illustrates the concept of this complexity class. The definition of the complexity class BPP states that more than half of all possible random strings $\underline{y}$ finally lead to an accepting configuration. If the PTM requires t time steps to solve a problem then there are 2^t possible paths.

The error can be made arbitrary small by repeating a probabilistic algorithm several times. The answer given most often is taken for granted. The probability that this answer is wrong decreases exponentially in the algorithm's error ϵ and the number of repetitions [NC00].

Besides the computational complexity classes P and BPP of efficiently solvable problems many other problems resisted to be solvable in polynomial time. These

[7] Upper bounds are indicated by the $\mathcal{O}$-notation: One writes $f = \mathcal{O}(g)$ for two functions f and g if there are constants $c > 0$ and $n_0 \in \mathbb{N}$ so that $f(n) \leq c \cdot g(n)$ for $n \geq n_0$. A lower bound $f = \Omega(g)$ denotes that there are constants $c > 0$ and $n_0 \in \mathbb{N}$ so that $f(n) \geq c \cdot g(n)$ for $n \geq n_0$. If $f = \mathcal{O}(g)$ and $f = \Omega(g)$ one writes $f = \Theta(g)$.

Complexity Class **BPP**
Let ϵ be a constant so that $0 \leq \epsilon < 1/2$. A language L belongs to complexity class BPP if there exists a PTM M so that for any input $\underline{x} \in \Sigma^*$ and $\underline{y}$ a string of random bits one of the following two conditions holds: i) $\underline{x} \in \mathrm{L} \Rightarrow$ the probability that M accepts $(\underline{x}, \underline{y})$ after $t = poly(\lvert\underline{x}\rvert)$ time steps is greater or equal to $1 - \epsilon$. ii) $\underline{x} \notin \mathrm{L} \Rightarrow$ the probability that M accepts $(\underline{x}, \underline{y})$ after $t = poly(\lvert\underline{x}\rvert)$ time steps is less or equal to ϵ.

Table 3.2: Definition of the complexity class BPP.

problems are largely optimization problems like the "traveling salesman problem" [FH02]. It was believed that there are no efficient algorithms that solve these algorithms. The theory of NP-completeness provided evidence in this belief.

The complexity class NP is defined in Tab. 3.3. If L∈NP then the configuration tree has only a small number of accepting paths. A language A is said to be NP-complete when $A \in$ NP and when all other languages L $\in$ NP can be transformed to A by a polynomial time TM.

Complexity Class **NP**
A language L belongs to complexity class NP if there exists a PTM M so that for any input $\underline{x} \in \Sigma^*$ and $\underline{y}$ a string of random bits one of the following two conditions holds: i) $\underline{x} \in \mathrm{L} \Rightarrow \exists \underline{y} \in \Sigma^*$ so that M accepts $(\underline{x}, \underline{y})$ after $t = poly(\lvert\underline{x}\rvert)$ time steps. ii) $\underline{x} \notin \mathrm{L} \Rightarrow \forall \underline{y} \in \Sigma^*$ M does not accept $(\underline{x}, \underline{y})$ after $t = poly(\lvert\underline{x}\rvert)$ time steps.

Table 3.3: Definition of the complexity class NP.

A proof of NP-completeness has come to signify the intractability of a problem [FH02]. In this case one usually tries other ways to solve the problem, e.g., approximation algorithms. Nevertheless, it is still not known whether problems in NP are really more difficult to solve than those in P [Gru00].

The quantum analogon to the classical complexity classes can be defined by replacing the classical TMs in the definitions above by QTMs. Bernstein *et al.* introduced the notation EQP to denote the quantum analogon to the classical complexity class P and the notation BQP to denote the quantum analogon to BPP [BV97]. Additionally,

they proved the following relations:[8]

$$\mathrm{P} \subseteq \mathrm{EQP} \subseteq \mathrm{BQP} \subseteq \mathrm{PSPACE}, \quad \mathrm{BPP} \subseteq \mathrm{BQP} \subseteq \mathrm{PSPACE}. \tag{3.4}$$

In classical complexity theory the following inclusions are known [Gru00]:

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE}. \tag{3.5}$$

It is still an open problem whether the inclusions in Eq. (3.5) are proper [Gru00]. Therefore, there is no possibility to give a mathematical proof whether BPP≠BQP unless one solves the major open problem P$\stackrel{?}{=}$PSPACE of complexity theory. Hence, one does not know if QTMs are more powerful than their classical pendants.

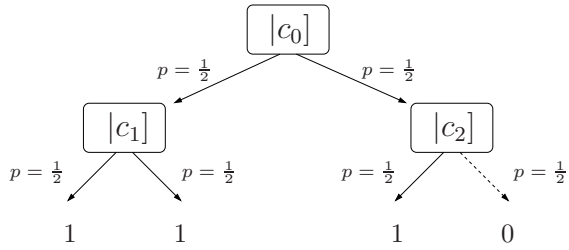

Figure 3.4: Configuration tree of a PTM: Each node represents an actual configuration $|c_i]$. Solid lines correspond to paths that lead to an accepting configuration.

3.1.5 Relativized Computational Complexity Classes

The result of Deutsch *et al.* [DJ92] was used by Berthiaume *et al.* to show that there exists an *oracle* relative to which a QA can solve a problem more efficiently than any classical deterministic algorithm [BB92, BB94]. To explore further what this means we have to introduce the concept of the *oracle TM* (OTM) [BB94, BGS75].

An OTM is a TM that has access to an oracle. The TM writes the oracle's input to an oracle tape. Then the oracle is executed within a single time step. It writes the corresponding output onto its tape. The output of the oracle is used by the TM for further calculations.

The name oracle has its origin in the fact that its internal functionality is assumed to be inaccessible.

If a TM accepts languages L of the complexity class B (L ∈ B), then the complexity class of languages accepted by the corresponding OTM whose oracle accepts languages L' with L' ∈ X, is called B^{X}. Such complexity classes are called *relativised complexity classes* [BGS75].

Berthiaume *et al.* demonstrated that the result of Deutsch *et al.* [DJ92] can be used to construct oracles X so that $\mathrm{P}^{\mathrm{X}} \subset \mathrm{EQP}^{\mathrm{X}}$ [BB92]. Relations between relativised complexity classes can be used to gain more insight into the relation between the corresponding unrelativised complexity classes [Gru00].

[8] The classical complexity class PSPACE denotes all formal languages L that are accepted by a classical TM that requires polynomial space (space = $poly(|\underline{x}|)$ for $\underline{x} \in$ L) and an arbitrary amount of time.

3.2 The Blackbox Model of Computation

As noted in the end of Sec. 3.1.4 it is difficult to decide whether the inclusions of Eq. (3.4) are proper. It is reasonable to investigate simpler models of computation because a better understanding of such easier models might ease the investigation of the original ones. The *decision tree* is perhaps the simplest model of computation [BdW02].

3.2.1 Decision Trees

In this model of computation one is given a blackbox $X = (x_0, x_1, \ldots, x_{N-1})$ containing N Boolean variables x_i with $i \in \{0, 1, \ldots, N-1\}$. In the most simplest form a query asks for the value of the variable x_i. On input i the box returns the corresponding value x_i. The goal is to calculate a property of the blackbox X which is represented by the Boolean function $f(X)$ using as few queries as possible. This number of queries is called *decision tree complexity*. The name decision tree is due to the fact that the algorithm is adaptive. This indicates that the kth query may depend on the outcome of the $k-1$ previous queries. The algorithm can be described by a binary tree (see Fig. 3.5).

One is only interested in the number of blackbox queries and not in the amount of additional processing between them. Also, the costs of a blackbox query are not considered. Therefore, such a query is related to the oracle call of a OTM, which is taken to be executed within a single time-step. Indeed, blackbox calls are also called oracle calls.

Analogous to Sec. 3.1.1 - 3.1.3 one defines deterministic decision trees, randomized decision trees and quantum decision trees. We now state the definitions and results of Buhrman *et al.* [BdW02].

A *deterministic decision tree* (see Fig. 3.5) is a rooted binary tree. Each node of this tree is labeled with a variable x_i. Let the root node be labeled with x_0. If the corresponding query returns 0 one has to recursively evaluate the left subtree, otherwise the right subtree. The output of the tree is the binary value (0 or 1) of the leaf that is finally reached. One says that the decision tree computes the property f when its output is equal to $f(X)$ for all $X \in \{0,1\}^N$. The depth of the tree is the worst-case number of calls necessary to reach the final leaf. There are many trees computing the same property f. Therefore, one defines the decision tree complexity $\mathrm{D}(f)$ to be the depth of an optimal (minimal-depth) decision tree that computes f.

A *randomized decision tree* is a deterministic one modified by additional nodes that, dependent on the outcome of a coin flip, call the left or right subtree. Now the input $X \in \{0,1\}^N$ no longer determines with certainty which leaf of the tree is finally reached. The complexity $\mathrm{R}_2(f)$ is the worst-case number of queries for the worst-case outcome of coin flips of a minimal-depth probabilistic decision tree. The outcome on input X equals $f(X)$ with a probability of at least $1-\epsilon$, with $0 \leq \epsilon < 1/2$.

A *quantum decision tree* (QDT) can be described by the unitary transformation $\boldsymbol{A}$ [BdW02]:

$$\boldsymbol{A} = \boldsymbol{U}_T \cdot \boldsymbol{O} \cdot \boldsymbol{U}_{T-1} \cdot \boldsymbol{O} \cdot \boldsymbol{U}_{T-2} \ldots \boldsymbol{O} \cdot \boldsymbol{U}_2 \cdot \boldsymbol{O} \cdot \boldsymbol{U}_1. \tag{3.6}$$

Here $\boldsymbol{U}_i$ is a fixed unitary transformation that does not depend on the input $X \in \{0,1\}^N$. The gate $\boldsymbol{O}$ denotes an oracle call.

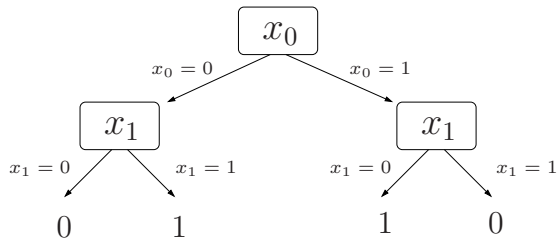

Figure 3.5: Deterministic decision tree that computes a property called parity of the blackbox $X = (x_0, x_1)$. The parity can be represented by the Boolean function $f(X) = x_0 \oplus x_1$.

The general form of the quantum state $|\psi\rangle$ of a QDT is $|\psi\rangle = |\underline{i}\rangle|b\rangle|\underline{h}\rangle|a\rangle$. The leftmost register stores the index i of the variable x_i that is to be queried next. The register $|b\rangle$ with $b \in \{0,1\}$ stores the output $|b \oplus x_i\rangle$ of the query. $|\underline{h}\rangle$ with $h \in \mathbb{N}$ is used to store the actual path (history) of the computation. The quantum state $|a\rangle$, with $a \in \{0,1\}$, represents the result of the computation.

To see how this works we start with the initial state $|\psi_0\rangle = |\underline{0}\rangle|0\rangle|\underline{0}\rangle|0\rangle$. Let the first unitary operation $\boldsymbol{U}_1$ map this state to the state $|\underline{i}\rangle|0\rangle|\underline{0}\rangle|0\rangle$. Applying the oracle gate $\boldsymbol{O}$ to this state returns $|\underline{i}\rangle|x_i\rangle|\underline{0}\rangle|0\rangle$. The next unitary operation U_1 maps this state to $|\underline{j}\rangle|0\rangle|\underline{h}_{i,x_i}\rangle|0\rangle$. The history $\underline{h}_{i,x_i}$ encodes the query i and the corresponding outcome x_i. Then the oracle is called for the value x_j and so on. Finally the answer of the algorithm is encoded into the rightmost qubit.

A QDT is not a tree anymore, nevertheless it can be used to simulate classical decision trees [BdW02]. Any T-query deterministic decision tree as well as any T-query randomized decision tree can be simulated by a T-query QDT with the same error probability.

The number of oracle calls is used to define the complexity of the QDT. If the QDT exactly computes the property $f(X)$ for all $X \in \{0,1\}^N$ then the QDT complexity is denoted by $\mathrm{Q_E}(f)$, here the index E stands for "exact". If the error in computing $f(X)$ is double bounded then the QDT complexity is denoted by $\mathrm{Q}_2(f)$.[9] For blackboxes $X \in \{0,1\}^N$ one gets [BBC+01]:

$$\mathrm{Q}_2(f) \leq \mathrm{R}_2(f) \leq \mathrm{D}(f) \leq N, \qquad \mathrm{Q}_2(f) \leq \mathrm{Q_E}(f) \leq \mathrm{D}(f) \leq N.$$

3.2.2 Quantum Lower Bounds by Polynomials

To ease the investigation of the computational power of decision trees other measures for the complexity of Boolean properties $f(X)$ have been studied. One of these measures is the degree of *approximating polynomials*. This approach allows to derive lower bounds on the number of oracle calls for *total Boolean functions* [BBC+01].

Total Boolean functions $f(X)$ are defined on all inputs $X \in \{0,1\}^N$ which is in contrast to *partial Boolean functions* that are only defined for subsets of blackboxes $X \in \{0,1\}^N$.

Now we define the *representing polynomial* of a Boolean function f:

[9] An error ϵ is called double bounded when the outcome on input X equals $f(X)$ with a probability of at least $1 - \epsilon$ with $0 \leq \epsilon < 1/2$.

- A polynomial $poly : \mathbb{R}^N \to \mathbb{R}$ represents f when $poly(X) = f(X)$ for all $X \in \{0,1\}^N$. This polynomial is unique.

The representing polynomial is a multilinear polynomial because x_i^2 equals x_i for $x_i \in \{0,1\}$. The degree $\deg(f)$ of f denotes the degree of the multilinear polynomial that represents f.

- A polynomial $poly : \mathbb{R}^N \to \mathbb{R}$ approximates f when $|poly(X) - f(X)| \leq \epsilon$ for all $X \in \{0,1\}^N$ and $0 \leq \epsilon < 1/2$.

The approximating degree $\widetilde{\deg}(f)$ is the minimum degree among all polynomials that approximate f.

Now we state an important result proven by Beals *et al.* [BBC+01]:

$$\mathrm{Q}_2(f) \geq \widetilde{\deg}(f)/2, \quad \mathrm{Q}_{\mathrm{E}}(f) \geq \deg(f)/2. \tag{3.7}$$

Proof:

Let the QDT start with the initial state $|\underline{0}\rangle|0\rangle|\underline{0}\rangle$ and apply a unitary operation $\mathbf{U}_0$ to it:

$$\boldsymbol{U}_0|\underline{0}\rangle|0\rangle|\underline{0}\rangle = \alpha|\underline{i}\rangle|0\rangle|\underline{z}\rangle + \beta|\underline{i}\rangle|1\rangle|\underline{z}\rangle.$$

For the sake of the argument we concatenated the history register and the output register into a single register. The oracle gate $\boldsymbol{O}$ maps $|\underline{i}\rangle|b\rangle|\underline{z}\rangle$ to $|\underline{i}\rangle|b \oplus x_i\rangle|\underline{z}\rangle$. One gets:

$$\begin{aligned}\boldsymbol{O}\left(\alpha|\underline{i}\rangle|0\rangle|\underline{z}\rangle + \beta|\underline{i}\rangle|1\rangle|\underline{z}\rangle\right) = \\ \underbrace{[(1-x_i)\alpha + x_i\beta]}_{\alpha'(x_i)}|\underline{i}\rangle|0\rangle|\underline{z}\rangle + \underbrace{[(1-x_i)\beta + x_i\alpha]}_{\beta'(x_i)}|\underline{i}\rangle|1\rangle|\underline{z}\rangle.\end{aligned} \tag{3.8}$$

After the first oracle query the probability amplitudes of the corresponding states are described by polynomials of degree 1 over the variable x_i. The unitary transformation $\boldsymbol{U}_1$ maps $|\underline{i}\rangle$ to $|\underline{j}\rangle$, $\alpha'(x_i)$ to $\alpha''(x_i)$ and $\beta'(x_i)$ to $\beta''(x_i)$. $\alpha''(x_i)$ and $\beta''(x_i)$ are linear combinations of $\alpha'(x_i)$ and $\beta'(x_i)$ and thus remain to be polynomials of degree 1. Now one applies the oracle gate again. Analogous to Eq. (3.8) with α and β replaced by $\alpha''(x_i)$ and $\beta''(x_i)$ as well as $|\underline{i}\rangle$ replaced by $|\underline{j}\rangle$ one obtains a polynomial over x_j and x_i of degree ≤ 2. Repeating this procedure for T queries one is left with a polynomial of degree $\leq$ T. Calculation of the measurement probability using these probability amplitudes returns polynomials of degree $\leq$ 2T. Hence, T oracle queries are sufficient to compute an approximating polynomial of degree 2T as stated by Eq. (3.7) □

The degree of representing and approximating polynomials can also be used to estimate the decision tree complexity of classical decision trees. Using the lower bounds for classical decision trees one gets [BBC+01]:

- If a QDT computes a total Boolean function f with bounded-error probability by making T oracle calls then there is a classical deterministic decision tree that computes f exactly making at most $O(T^6)$ queries.

- If a QDT computes a total Boolean function f exactly by making T oracle calls then there exists a classical deterministic decision tree that computes f exactly making at most $O(T^4)$ queries.

We have evolved algorithms to solve the parity-problem, therefore we now present the corresponding tight bounds (see Eq. (3.9)) proven by Beals *et al.* [BBC+01].

In the parity problem one has to decide if the blackbox $X \in \{0,1\}^N$ contains an even or an odd number of 1s. The parity of X can be represented by the total Boolean function $f(X) = x_{N-1} \oplus x_{N-2} \oplus \cdots \oplus x_0$. One gets:[10]

$$\mathrm{Q_E}(f) = \mathrm{Q}_0(f) = \mathrm{Q}_2(f) = N/2. \tag{3.9}$$

The results stated in this section show that QAs do at most provide polynomial speed-ups in the computation of total Boolean functions. Indeed, QAs that show an exponential speed-up are those with a certain promise on the oracles like in DJ's problem [DJ92]. There the promise on the blackboxes is very restrictive. Therefore, even a probabilistic classical algorithm can efficiently solve this problem.

The results on the quantum speed-up in computing total Boolean functions is disappointing. But the considerations above are only based on worst-case scenarios. In this context one also speaks of the *worst-case query complexity* of algorithms. In "real life" it is also interesting to consider the *average-case query complexity* where algorithms are allowed to use different numbers of blackbox calls for different blackboxes.

The average-case quantum query complexity of total Boolean functions was investigated by Ambainis *et al.* whose results show that there are problems on which QAs provide an exponential speed-up [AdW01]. Therefore, the number of worst-case inputs $X \in \{0,1\}^N$ for QDTs can be exponentially smaller than the number of worst-case inputs for probabilistic decision trees.

The authors further showed that any QDT that solves the parity problem would still require at least $\Omega(N)$ oracle calls on average. Hence, even in the average-case scenario no more than a linear quantum speed-up is possible in solving the parity problem.

3.3 Circuit Model of Computation

In this section we will introduce the circuit model of computation following the lines of Hirvensalo [Hir01]. Although this model is not computational equivalent to the TM it is frequently used in the study of QAs.

Any Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be decomposed into a set of elementary logical operations. This can be seen by employing the disjunctive normal form: A Boolean function $f(x_{n-1}, x_{n-2}, \ldots, x_0)$ of the Boolean variables $x_i \in \{0,1\}$ is decomposed into a disjunction of clauses each of which is a conjunction of some Boolean variables x_i or negations thereof. The term disjunction denotes the logical operation OR abbreviated by "$\vee$", the term conjunction denotes the logical operation AND abbreviated by "$\wedge$". Together with the negation symbol "$\neg$" the Boolean

[10] Here we used $\mathrm{Q}_0(f)$ to denote an additional complexity measure of a QDT not stated yet: Imagine that the QDT is allowed to answer inconclusive with a probability of at most 1/2. If it does not answer inconclusive the answer has to be right with certainty. The complexity measure $\mathrm{Q}_0(f)$ denotes the number of oracle queries of such a QDT.

function $f(x_1, x_0)$ defined by $f(0,0) = 0$, $f(0,1) = 1$, $f(1,0) = 1$ and $f(1,1) = 0$ is represented by:[11]

$$f(x_1, x_0) = (\neg x_0 \wedge x_1) \vee (x_0 \wedge \neg x_1). \tag{3.10}$$

Each of the elementary logical gates $(\vee, \wedge, \neg)$ only applies to a small subset of inputs. Additionally, the rules that specify the actions of the logical gates are given finitely. This can be seen by their corresponding truth tables (see Tab. 3.4). A physical system that implements such a decomposition operates by finite means and therefore provides an alternative approach to computability.

$\vee(a,b)$	a	b
0	0	0
1	0	1
1	1	0
1	1	1

$\wedge(a,b)$	a	b
0	0	0
0	0	1
0	1	0
1	1	1

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

$\neg a$	a
1	0
0	1

Table 3.4: Truth tables for some elementary logical operations.

The decomposition of a Boolean function can be represented by a *Boolean circuit* (see Fig. 3.6). This is an acyclic, directed graph whose nodes are labeled either with the input variables, the output variables, or the elementary logical gates $(\vee, \wedge, \neg)$. The arrows of the graph that connect the nodes are called wires.

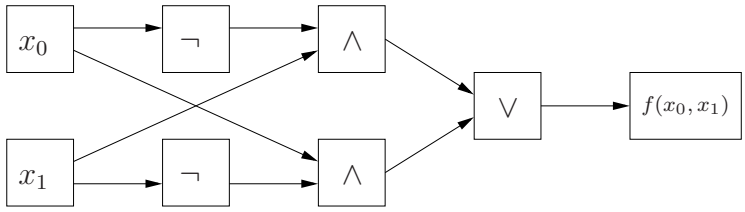

Figure 3.6: Boolean Circuit that computes the Boolean function $f(x_1, x_0) = (\neg x_0 \wedge x_1) \vee (x_0 \wedge \neg x_1)$. The nodes represent either input variables, output variables or the elementary logical gates $(\vee, \wedge, \neg)$.

As mentioned above any Boolean function f whose truth table is known can be decomposed into elementary logical operations. Therefore, it can be represented by a Boolean circuit. Usually one does not know the truth table in advance, thus one is interested in an algorithm that, on input n, returns the circuit C_n that calculates the Boolean function f_n. Here we used f_n to denote that the number of arguments is restricted to n bits.

[11] This representation of f is constructed by taking those assignments a_1, a_0 that fulfill $f(a_1, a_0) = 1$ to build clauses $c_i(x_1, x_0)$ so that $c_i(a_1, a_0) = 1$. As stated above a clause has to be a conjunction of the variables x_i or $\neg x_i$. A clause that fulfills $c_1(0,1) = 1$ is given by $c_1(x_1, x_0) = \neg x_1 \wedge x_0$. A clause that fulfills $c_2(1,0) = 1$ is given by $c_2(x_1, x_0) = x_1 \wedge \neg x_0$. The disjunction $c_1 \vee c_2$ represents the Boolean function $f(x_1, x_0)$ defined above. Using this kind of construction any Boolean function can be decomposed into elementary logical gates.

A *circuit family* is an infinite sequence $C_0, C_1,...,C_n,...$ of individual Boolean circuits C_n. A family of Boolean circuits computes f. Hence, it can be used to recognize a language $\mathrm{L} \subset \{0,1\}^*$.

The number c_n of elementary gates required by the Boolean circuit C_n provides a measure of its efficiency. This number of elementary gates is called the *size* of a circuit.

A family of Boolean circuits C_n whose design is generated by a TM in polynomial time $t = \mathcal{O}\,(\,poly\,(c_n)\,)$ is called a *uniform circuit family.*

As stated by I. Wegener a TM with running time $t(n)$ can be simulated by a uniform circuit C_n of size $\mathcal{O}\,(\,t(n) \cdot \log\,(t(n))\,)$ [Weg03]. Thus, any language L with $\mathrm{L} \in \mathrm{P}$ has circuits of polynomial size. Nevertheless, not all languages recognized by polynomial size circuits are in P [Pap94]:

- A language L has uniform polynomial size circuits if and only if $\mathrm{L} \in \mathrm{P}$. All languages $\mathrm{L} \in \mathrm{BPP}$ have polynomial size circuits but these circuits have not to be uniform!

The concept of Boolean circuits can easily be extended to functions $f : \{0,1\}^n \to \{0,1\}^m$ by using m functions each of which computes a single bit.

The logical gates $\vee$ and $\wedge$ are irreversible operations. It is well known that these logical gates can be efficiently simulated by a reversible logical gate known as Toffoli gate T which is defined by [NC00]:

$$\mathrm{T}(a,b,c) = (a,b,c \oplus a \cdot b) \quad \text{with } a,b,c \in \{0,1\}.$$

This gate can be realized on quantum physical devices. Therefore, every function that is computable by a Boolean circuit can be computed by a *quantum circuit* (QC).

In the quantum regime the Toffoli gate is not the most elementary gate. Any arbitrary unitary transformation can be decomposed in unitary gates acting on one and two qubits [NC00]. With the Euler-decomposition any arbitrary one-qubit operation $\boldsymbol{U}$ can be decomposed by:

$$\boldsymbol{U} = e^{\imath\alpha}\boldsymbol{R}_x(\beta)\boldsymbol{R}_y(\gamma)\boldsymbol{R}_x(\delta), \quad \alpha,\beta,\gamma,\delta \in \mathbb{R}. \tag{3.11}$$

In contrast to classical logical operations quantum gates depend on continuous parameters $\alpha \in \mathbb{R}$. These parameters have to be discretized, hence one only can approximate unitary operations. Any unitary operation on one qubit can be approximated arbitrarily close by a discrete set of the gates $\boldsymbol{R}_x(2\cdot\alpha_i)$, $\boldsymbol{R}_y(2\cdot\alpha_i)$ and $\boldsymbol{R}_z(2\cdot\alpha_i)$ [NC00]. The discrete value α_i is calculated by $\alpha_i = i \cdot \frac{2\pi}{\xi}$ with $i \in \{0,1,\ldots\xi-1\}$. ξ is chosen dependent on the desired accuracy of the approximation.

Additional to this complete set of elementary one-qubit gates we introduce the Hadamard gate $\boldsymbol{H}$ because it is frequently used in the quantum computation literature:

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

A single discrete two-qubit operation called $\boldsymbol{CNOT}$-gate together with the discrete set of one-qubit gates from above is sufficient to provide a complete set of unitary operations. This set of quantum gates is capable of approximating any arbitrary

unitary operation up to the desired accuracy [NC00]. The ***CNOT***-gate is defined by its action on a two-qubit state:

$$\boldsymbol{CNOT}\,|x\rangle|y\rangle = |x\rangle|y \oplus x\rangle \;\text{ with }\; x, y \in \{0, 1\}. \tag{3.12}$$

It is also possible to define a classical CNOT operation for the Boolean inputs (x, y) by $\mathrm{CNOT}(x, y) = (x, y \oplus x)$. This operation can be used to perform copy operations: $\mathrm{CNOT}(x, 0) = (x, x)$. But this interpretation does not hold in the quantum regime where the no-cloning theorem does not allow for copying an arbitrary quantum state unless it is one of the two computational basis states $|0\rangle$ or $|1\rangle$ [NC00].

A QC can be represented by a directed acyclic graph whose nodes are labeled by quantum operations. Due to reversibility and the no-cloning theorem any elementary gate has as many inputs as outputs. Using the convention that such a graph is to be read from the left to the right, the arrows used in the representation of Boolean circuits can be replaced by straight lines.

The complexity of QCs is the number of elementary quantum gates necessary to approximate a unitary matrix. How many elementary quantum gates are necessary on average to approximate an arbitrary unitary $2^n \times 2^n$ matrix of an n-qubit system up to an error ϵ?[12] Nielsen and Chuang show that the average number m of such elementary quantum gates is lower bounded by $m = \Omega\left(\frac{2^n \log(1/\epsilon)}{\log(n)}\right)$ [NC00].

A similar argument holds for classical Boolean circuits: The average number of elementary classical logical gates needed to realize a Boolean function with n input bits is lower bounded by $\Omega(2^n/n)$ [Sha49].

These results show that the representation of a Boolean function by a Boolean circuit or a QC is generically hard.

Nishimura *et al.* define uniform QC families $Q_0, Q_1, \ldots, Q_n, \ldots$ to be circuit families whose design is returned by a DTM in polynomial time $t = \mathcal{O}(\,poly(q_n)\,)$, where q_n denotes the size of the QC Q_n [NO02]. Then they define the following complexity classes of languages recognizable by uniform QC families: EUPQC (**e**xact **u**niform **p**olynomial **q**uantum **c**ircuit), ZUPQC (**z**erro-error **u**niform **p**olynomial **q**uantum **c**ircuit) and BUPQC (**b**ounded-error **u**niform **p**olynomial **q**uantum **c**ircuit). They prove:

$$\mathrm{EUPQC} \subset \mathrm{ZUPQC} \subset \mathrm{BUPQC}.$$

They further prove that languages that are efficiently recognizable by Monte Carlo type uniform QC families are also efficiently recognized by Monte Carlo type QTMs, and vice versa:[13]

$$\mathrm{BQP} = \mathrm{BUPQC}.$$

Before we present some blackbox QAs we have to add the oracle gate to the set of elementary gates. The internal workings of the oracle gates is not of interest and it is assumed that the oracle gate performs its operation in a single time step. The QDT defined in Eq. (3.6) can be realized by a QC whose gates $\boldsymbol{U}_i$ but the oracle gate are decomposed into elementary quantum gates. This makes it possible to compare

[12] The error $\epsilon(\boldsymbol{U}, \boldsymbol{V})$ of a unitary matrix $\boldsymbol{V}$ in approximating the unitary matrix $\boldsymbol{U}$ is defined by $\epsilon(\boldsymbol{U}, \boldsymbol{V}) \equiv \max_{|\psi\rangle} ||(\boldsymbol{U} - \boldsymbol{V}) \cdot |\psi\rangle||$.

[13] The term *Monte Carlo type* denotes computation devices that recognize languages with a bounded error probability. With this notation the classical complexity class BPP introduced in Sec. 3.1.4 is the set of languages that are recognized by Monte Carlo type TMs.

different blackbox QAs not only with regard to their query complexity but also to their size.

3.4 Quantum Algorithms

The blackbox model of computation (see Sec. 3.2) provides a natural description of a large class of QAs [BBC+01]. The problem proposed by Deutsch (see Tab. 3.5) can be expressed adequately using this approach [Deu85].

Deutsch's Problem
One is given a blackbox $X = (x_0, x_1)$ of two Boolean variables x_0 and x_1. On input $i \in \{0, 1\}$ the blackbox returns the corresponding value x_i. Calculate the Boolean function $f(X) = x_0 \oplus x_1$ of X using as few blackbox queries as possible.

Table 3.5: Definition of Deutsch's Problem

In Deutsch's problem one calculates the parity of the blackbox $X = (x_0, x_1)$. A QC can do this with less oracle calls than any classical circuit (see Sec. 3.2.2).

Fig. 3.7a depicts how a physical experiment is used to solve the parity problem for a blackbox X. Given are two tubes, each of which can be modified so that an incoming light either suffers a phase shift of π or not. Therefore, each tube represents a Boolean variable $x_i \in \{0, 1\}$. Two tubes represent a blackbox $X = (x_0, x_1)$. Deutsch's problem can now be modeled by assigning the value 1 to a tube that performs a phase shift to the incoming light beam and the value 0 to a tube that does not alter the light beam.

To calculate the parity using the classical approach to computation (no superpositions allowed over queries) one has to query each tube independently by a light beam in order to see whether a phase shift occurs. If both beams are altered, or if no beam is altered at all, then parity is equal to 0, otherwise parity is equal to 1. Hence, one needs two calls to solve the parity problem.

Fig. 3.7b shows how the parity problem can be solved by only one oracle call. The blackbox is called by sending a light beam. Now a single beam is sufficient to solve the parity problem. The beam is split into a superposition of two light beams each of which is used to query a single blackbox element. The final beam splitter joins both beams and therefore detector 1 measures the photon if and only if the parity of the blackbox is 0.[14] The speed-up in the number of oracle calls is due to the appropriate usage of superpositions and interference.

In describing the functionality of this experiment we didn't require quantum physical concepts. If one uses a unary encoding of computational states then this speed-up can also be explained in terms of classical physics.

In both experiments the two different paths of the light beam(s) are used to encode the value $i \in \{0, 1\}$. Therefore, one speaks of a unary encoding. This approach

[14] If two light beams show a phase difference of π then they interfere destructively. Then the detector does not measure anything.

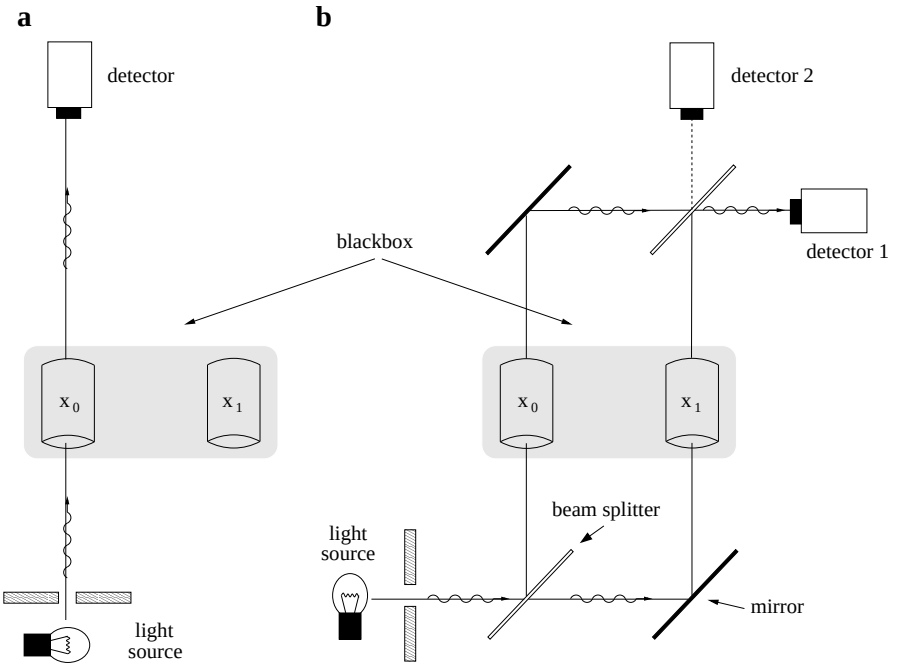

Figure 3.7: This figure shows the two approaches a and b to call the blackbox $X = (x_0, x_1)$. The blackbox is represented by two tubes (gray area). Each tube can be modified so that an incoming light beam either suffers a phase shift of π or not.

requires 128 paths to encode $i \in \{0, \ldots, 127\}$. It is better to use binary encoding with two internal states of the light beam, e.g., the polarization. Binary encoding reduces the number of different light paths necessary to encode $i \in \{0, \ldots, 127\}$ to $\ln_2(128) = 7$ paths.

S. Lloyd notes that a unary encoding is inefficient since it requires exponentially more resources than usage of a binary encoding would [Llo00].

Binary encoding and superpositions usually lead to entanglement which is a genuine quantum phenomenon (see Sec. 2.6): Consider a blackbox call represented by an oracle gate $\boldsymbol{O}$ whose action onto the state $|\underline{i}\rangle = |i_1 i_0\rangle$, with $i_1, i_0 \in \{0, 1\}$, is described by:[15]

$$\boldsymbol{O}|\underline{i}\rangle = (-1)^{x_i}|\underline{i}\rangle \quad \text{with} \quad i \in \{0, 1, 2, 3\}.$$

Let the blackbox $X = (x_0, x_1, x_2, x_3) = (0, 1, 0, 0)$ be called by the following product state $|\psi\rangle$:

$$|\psi\rangle = \frac{1}{2}\left(|0\rangle + |1\rangle\right) \otimes \left(|0\rangle + |1\rangle\right) = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right).$$

[15] To motivate this definition of the oracle gate consider the effect of the blackbox call in Fig. 3.7b. In its classical as well as in its quantum mechanical treatment the light beam is described by a sine wave. A tube thus inverts the sine wave if it shifts its phase by π. If one uses $|0\rangle$ ($|1\rangle$) to denote the light beam in the lower (upper) path of the interferometer the effect of a tube that shifts the phase is described by $|0\rangle \to -|0\rangle$ ($|1\rangle \to -|1\rangle$).

After executing the blackbox call the resulting state looks like:

$$\boldsymbol{O}|\psi\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle + |11\rangle\right) \neq (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_0|0\rangle + \beta_0|1\rangle), \quad (3.13)$$

with $\alpha_1, \alpha_0, \beta_1, \beta_0 \in \mathbb{C}$. The resulting state is not decomposable into a tensor product. It is an entangled state and hence only realizable by a quantum system.

There may be problems whose blackboxes can be represented by oracle gates that do not map unentangled states to entangled ones. Indeed, D. Meyer [Mey00] proposes a QA that performs a "sophisticated" database search using such non-entangling oracle gates. Any conventional classical algorithm that uses the same oracle needs n queries. Meyer's QA solves the problem by only one oracle query.

In the next two sections we will present some QAs that are related to the algorithms we evolved using GP. We only evolved algorithms for decision problems, therefore we will restrict our discussion to this class of algorithms.

3.4.1 The Deutsch-Jozsa Problem

The DJ problem defined in Tab. 3.6 was the first one that showed an exponential gap in the number of oracle calls needed by an exact QA and those needed by any exact classical deterministic algorithm [DJ92]. Deutsch *et al.* emphasize that this problem could also be efficiently solved by a classical probabilistic algorithm [DJ92]. Simon's problem was the first one with an exponential gap in the number of oracle calls required by his QA and those required by any classical algorithms [Sim94]. This problem will be discussed in Sec. 3.4.2.

We now present a modified form of the DJ QA proposed by Cleve *et al.* [CEMM97]. This QA only requires one oracle call. The original DJ QA needs two calls. Another modification of the DJ QA is due to Collins *et al.* whose version does not require any ancillary qubits [CKH98].

The DJ problem
One is given a blackbox $X = (x_0, x_1, \ldots, x_{N-1})$ of N Boolean variables x_i so that on input $i \in \mathcal{I}$ with $\mathcal{I} = \{0, 1 \ldots, N-1\}$ the blackbox returns the corresponding value x_i. One is promised that either the blackbox X belongs to set A of blackboxes with $f(X) = 0$ or to set B of blackboxes with $f(X) = 1$: i) $X \in A \quad \Leftrightarrow \quad \forall i \in \mathcal{I} \; : \; x_i = c$ with $c \in \{0, 1\}$ ii) $X \in B \quad \Leftrightarrow \quad x_i = 0$ for exactly one half of all $i \in \mathcal{I}$ Decide which set X belongs to.

Table 3.6: Definition of the DJ problem.

The property of a blackbox has to be a partial Boolean function if a QA should provide an exponential speed-up in the number of oracle calls (see Sec. 3.2). Indeed,

this is the case with DJ's problem. The QA that solves this problem is only slightly faster than a classical probabilistic algorithm. Hence, a promise on the blackbox does not necessarily imply that a QA solves this problem exponentially faster than any classical algorithm.

Any deterministic decision tree would need at most $N/2+1$ queries to answer the DJ problem: If $N/2$ different queries return the same answer then it is still possible that the blackbox belongs to set B. The probability that such a worst-case occurs is exponentially small. Therefore, a probabilistic decision tree can solve this problem efficiently:

Proof:

A probabilistic decision tree needs at least two oracle calls to solve the problem: The probability that two randomly chosen queries return the same value for a blackbox $X \in B$ is $1/2$ for $N \gg 1$. □

A QA can solve this problem by only one oracle call:

Proof:

One starts in the initial state $|\psi_0\rangle = |\underline{0}\rangle$ of the n-qubit query register . This register is used to address the $N = 2^n$ different elements x_i of the blackbox. By applying a Hadamard gate $\boldsymbol{H}$ to each of the n qubits one gets a superposition over all query states:

$$|\psi_1\rangle = \boldsymbol{H}^{\otimes n}|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\underline{i}\rangle.$$

The oracle gate $\boldsymbol{O}$ is defined by:

$$\boldsymbol{O}|\underline{i}\rangle = (-1)^{x_i}|\underline{i}\rangle.$$

It is queried by this superposed state. One gets:

$$|\psi_2\rangle = \boldsymbol{O}|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{x_i}|\underline{i}\rangle.$$

Once again a Hadamard gate is applied to each qubit. Because of:

$$\boldsymbol{H}^{\otimes n}|\underline{i}\rangle = \frac{1}{\sqrt{2^n}} \sum_j (-1)^{\underline{i}\cdot\underline{j}}|\underline{j}\rangle, \tag{3.14}$$

where $\underline{i} \cdot \underline{j}$ is the bitwise inner product of i and j modulo 2, one gets:

$$|\psi_3\rangle = \boldsymbol{H}^{\otimes n}|\psi_2\rangle = \frac{1}{2^n} \sum_j \sum_i (-1)^{\underline{i}\cdot\underline{j}+x_i}|\underline{j}\rangle.$$

It is sufficient to calculate the probability amplitude a_0 of state $|\underline{0}\rangle$:

$$a_0 = \left(\frac{1}{2^n} \sum_{i=0}^{2^n-1} (-1)^{x_i}\right) = \begin{cases} \pm 1 & \text{if} \quad X \in A \\ 0 & \text{if} \quad X \in B \end{cases}.$$

A measurement of the final n-qubit state reveals exactly if $X \in A$ or $X \in B$. □

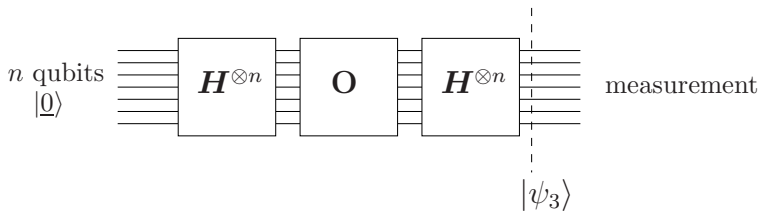

Figure 3.8: Schematic diagram of the QA that solves the DJ problem. This algorithm distinguishes between blackboxes $X \in A$ and blackboxes $X \in B$ by mapping the initial state $|\psi_0\rangle = |\underline{0}\rangle$ to the final state $|\psi_3\rangle = |\underline{0}\rangle$ if $X \in A$, otherwise the state $|\underline{0}\rangle$ has a probability amplitude of zero.

3.4.2 Simon's Problem

Simon's QA was the first QA that solved a problem exponentially faster than any classical algorithm [Sim94]. Drawing on this algorithm Shor developed his polynomial-time QAs for discrete logarithm and integer factoring [Sho94].

As well as DJ's problem Simon's problem can be stated using the blackbox model of computation. Up to now we only considered blackboxes $X = (x_0, x_1, \ldots, x_{N-1}) \in \{0,1\}^N$ with Boolean variables $x_i \in \{0,1\}$. To state Simon's problem we have to consider variables $x_i \in \{0,1\}^m$. For the n-qubit query $\underline{i}$ with $\underline{i} \in \{0,1\}^n$ the blackbox returns the m-bit integer $\underline{x_i}$. Simons' problem is defined in Tab. 3.7.

Simon's Problem
One is given a blackbox $X = (x_0, x_1, \ldots, x_{N-1})$ of $N = 2^n$ variables $\underline{x_i} \in \{0,1\}^m$ with $m \in \mathbb{N}$ so that on input $i \in \mathcal{I}$ with $\mathcal{I} = \{0,1,\ldots N-1\}$ the blackbox returns the integer x_i. We are promised that either the blackbox X belongs to set A of blackboxes with $f(X) = 0$ or to set B of blackboxes with $f(X) = 1$: i) $X \in A \;\Leftrightarrow\; (\forall i \neq j \text{ with } i,j \in \mathcal{I} : \; x_i = x_j \Leftrightarrow \underline{j} = \underline{i} \oplus \underline{s} \text{ with } \underline{s} \in \mathcal{I})$ ii) $X \in B \;\Leftrightarrow\; \big(\forall i,j \in \mathcal{I} : \; x_i = x_j \Leftrightarrow \underline{j} = \underline{i} \oplus \underline{0} = \underline{i}\big)$ Decide which set X belongs to. If $X \in A$ then determine $\underline{s}$.

Table 3.7: Definition of Simon's problem.

Before we present Simon's QA we will prove that any classical probabilistic algorithm that queries the oracle no more than $2^{n/4}$ times cannot correctly guess whether $X \in A$ or $X \in B$ with a probability greater than $(1/2) + 2^{-n/2}$. The following arguments are analogous to those presented by Simon [Sim94]:

Proof:

A classical algorithm solves this problem when it finds two values $i, j \in \mathcal{I}$ so that $x_i = x_j$. The probability that this happens will be denoted by δ. Using this definition the probability that a classical algorithm guesses correctly is given by $1/2 + \delta$. To calculate δ one first has to determine the number of pairs (x_i, x_j) that can be tested by k oracle queries. With two oracle queries only a single pair is tested ($\xi = 1$). With the next oracle query two additional pairs are tested ($\xi = 2$) and so on. Hence, $\sum_{\xi=1}^{k-1} \xi = (k-1)k/2$ pairs can be tested by k queries. We used the variable ξ to denote the new pairs that can be tested by each additional oracle query. To each tested pair (x_i, x_j) corresponds an integers s so that $\underline{i} = \underline{j} \oplus \underline{s}$. With k queries at most $(k-1)k/2$ different values for s are tested. If after k queries no pair (x_i, x_j) with $x_i = x_j$ was found then there are $2^n - (k-1)k/2$ untested values left for s. Now we calculate the probability p_k that after $k-1$ unsuccessful queries the k-th query tests the right value s: After $k-1$ unsuccessful queries there are still $2^n - (k-2)(k-1)/2$ untested values s. The k-th query tests $\xi = k-1$ new pairs out of these $2^n - (k-2)(k-1)/2$ candidates. One gets:

$$p_k = \frac{k-1}{2^n - (k-2)(k-1)/2}.$$

The total probability δ to find a pair (x_i, x_j) with $x_i = x_j$ by $k = 2^{n/4}$ queries is the sum over all conditional probabilities p_k. For legibility we introduce the constant $\mu = 2^{n/4}$:

$$\begin{aligned}\delta &= \sum_{k=2}^{\mu} p_k = \sum_{k=2}^{\mu} \left(\frac{k-1}{2^n - (k-1)(k-2)/2}\right) \leq \sum_{k=1}^{\mu} \left(\frac{k}{2^n - (\mu-1)(\mu-2)/2}\right) \\ &\leq \sum_{k=1}^{\mu} \frac{k}{\mu^4 - \mu^2} = \frac{\mu(\mu-1)/2}{\mu^4 - \mu^2} \leq \frac{1}{\mu^2} = 2^{-n/2}.\end{aligned}$$

Therefore, no classical algorithm can decide if $X \in A$ or $X \in B$ with a success probability greater than $1/2 + 2^{-n/2}$ by $2^{n/4}$ oracle queries. □

Simon solved this problem by $\mathcal{O}(n)$ repetitions of a QA (see Fig. 3.9) that calls the blackbox only once [Sim94]:

Proof:

The blackbox returns m-bit integers on n-bit queries. Therefore, the oracle-gate acts on a query register of n qubits and an output register of m qubits:

$$\boldsymbol{O}|\underline{i}\rangle|\underline{b}\rangle = |\underline{i}\rangle|\underline{b} \oplus \underline{x_i}\rangle.$$

In Fig. 3.9 the QA starts with the initial state $|\psi_0\rangle = |\underline{0}\rangle|\underline{0}\rangle$. By applying a Hadamard gate to each of the n qubits of the query register one gets:

$$|\psi_1\rangle = \left(\boldsymbol{H}^{\otimes n}|\underline{0}\rangle\right) \otimes |\underline{0}\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\underline{i}\rangle|\underline{0}\rangle.$$

Applying the oracle gate to this superposition returns:

$$|\psi_2\rangle = \boldsymbol{O}|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\underline{i}\rangle|\underline{x_i}\rangle.$$

Hadamard gates are performed on each of the n qubits of the input register:

$$|\psi_3\rangle = \left(\boldsymbol{H}^{\otimes n} \otimes \mathbf{1}^{\otimes m}\right)|\psi_2\rangle = \frac{1}{\sqrt{2}} \sum_{i=0}^{2^n-1} \left(\boldsymbol{H}^{\otimes n}|\underline{i}\rangle\right)|\underline{x_i}\rangle = \frac{1}{2^n} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} (-1)^{\underline{i}\cdot\underline{j}}|\underline{j}\rangle|\underline{x_i}\rangle.$$

The right side of the equation above follows because of Eq. (3.14).

If $X \in B$ then all values x_i are different. Therefore, the input register is in a superposition over the 2^n states $|\underline{j}\rangle$. An n-qubit measurement returns one of these states with a probability of $1/2^n$. For $X \in A$ one can rewrite $|\psi_3\rangle$:

$$|\psi_3\rangle = \frac{1}{2^{n-1}} \sum_i \sum_{j=0}^{2^n-1} \left[(-1)^{\underline{i}\cdot\underline{j}} + (-1)^{(\underline{i}\oplus\underline{s})\cdot\underline{j}}\right]|\underline{j}\rangle|\underline{x_i}\rangle.$$

In this equation one sums over all indices i with different values of x_i. The condition $(-1)^{\underline{i}\cdot\underline{j}} + (-1)^{(\underline{i}\oplus\underline{s})\cdot\underline{j}} \neq 0$ is equivalent to the condition $\underline{i}\cdot\underline{j} = (\underline{i}\oplus\underline{s})\cdot\underline{j}$. This is equivalent to $\underline{s}\cdot\underline{j} = 0$. One gets:

$$|\psi_3\rangle = \frac{1}{2^{n-2}} \sum_i \sum_{\underline{j}:\, \underline{j}\cdot\underline{s}=0} (-1)^{\underline{i}\cdot\underline{j}}|\underline{j}\rangle|\underline{x_i}\rangle.$$

One sums over indices j that fulfill $\underline{j}\cdot\underline{s} = 0$. Measuring the input register hence returns a value $\underline{j}$ so that $\underline{j}\cdot\underline{s} = 0$. By at least n repetitions of Simon's quantum algorithm one receives n linearly independent strings $\underline{j}$. These strings are used to calculate the binary string $\underline{s}$. For $X \in A$ one gets the binary string $\underline{s}$ one is interested in. For $X \in B$ the string $\underline{s}$ will be random. One can distinguish between $X \in A$ and $X \in B$ by querying the oracle for the value x_0 and x_s. If $x_0 = x_s$ then $X \in A$, else $X \in B$. □

Simon's problem can be solved by $\mathcal{O}(n)$ repetitions of the QA. Each repetition calls the blackbox only once. Therefore, the problem can be solved by $\mathcal{O}(n)$ oracle calls. Any classical algorithm requires an exponential number of oracle calls. Thus, there is an exponential gap in the number of calls needed by a QA and those needed by any classical algorithm.

3.4.3 Additional Remarks on Quantum Algorithms

Simon's problem can be solved efficiently by a QA because of its internal algebraic structure: One is promised that either all of the N elements of the blackbox have distinct values or that the N elements have $N/2$ distinct values. Two elements x_i and x_j are equal if and only if $\underline{j} = \underline{i} \oplus \underline{s}$.

A similar problem without such an internal algebraic structure is known under the name *collision problem*. Analogous to Simon's problem one is promised that either all

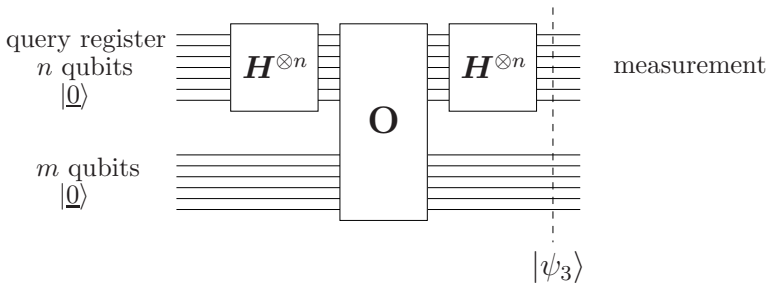

Figure 3.9: Schematic diagram of the QA that solves Simon's problem. This algorithm distinguishes between blackboxes $X \in A$ and blackboxes $X \in B$ by mapping the initial state $|\underline{0}\rangle$ of the query register to a superposition over the states $|\underline{j}\rangle$ with $\underline{s} \cdot \underline{j} = 0$ if $X \in A$. For $X \in B$ one gets a superposition over all states $j \in \mathcal{I}$.

of the N elements of the blackbox have distinct values or that the N elements have $N/2$ distinct values. In the latter case always two elements are equal. One has to decide which one of these two promises holds for a blackbox X.

Any QA that solves the collision problem needs at least $\Omega(\sqrt[3]{\frac{N}{2}})$ blackbox queries whereas a classical algorithm needs $\Theta(\sqrt[2]{\frac{N}{2}})$ queries [Shi92]. This indicates that a promise on the blackboxes is not sufficient to get an exponential quantum speed-up.

There exists a class of problems, known as *hidden subgroup problems*, whose internal structure allows for such a speed-up. Simon's problem is a special case of the hidden subgroup problem. Shor's factoring QA as well as his discrete logarithm QA exploit an internal hidden subgroup structure, too (see [NC00]).

4 Genetic Programming and How to Evolve Quantum Algorithms

The last chapter introduced the framework necessary to speak about computability, algorithms and circuits. Nevertheless, often it is a tedious task to find a circuit or algorithm that solves a posed problem. If it is possible to quantify the success of a candidate circuit/algorithm then it may be advantageous to use an automatic programming technique that "breeds" potential solutions by using general principles from genetics and evolutionary biology.

The automated programming technique used in this thesis to develop QCs and QAs is called GP. An overview of GP can be found in the book of Banzhaf *et al.* [BNKF98].

The usage of GP to aid the development of QCs was pioneered by Williams *et al.* who used this approach to decompose a given quantum transformation into a sequence of elementary quantum gates [WG98]. Spector *et al.* presented a GP system that is able to develop QCs without knowing the quantum transformation in advance [SBBS99]. Ever since several related strategies using GP to aid the development of QCs/QAs have been proposed and investigated by several authors [Rub01, LB03a, MCS04, Spe04].

Most of these approaches focused on the evolution of QCs like the two-bit and/or problem [BBS00, SK06], teleportation circuits [Rub01], circuits solving two-qubit instances of DJ's and Grover's problem [SBBS99, Spe04] or circuits to find the maximum of a permutation function [MCS04]. Some of these approaches were also used to develop scalable QCs (uniform circuit families) for problems like the majority-on problem [SBBS99], 1-SAT problems [LB03a] or finding the maximum of permutation functions [MCS04].

For the majority-on problem no better-than-classical algorithm was found. The algorithm found for the 1-SAT problem made it possible to optimize Hogg's QA with respect to non-oracle gate operations required. Unfortunately, the algorithm found to solve the maximum-finding problem only boosts the probability to measure the correct value by a factor of two in comparison to a classical algorithm that samples the function at random. Therefore, this QA is ineffective because the number of sample points grows exponentially in the number of bits.

In this chapter we will show how GP can be incorporated to provide a useful tool in designing formerly unknown better-than-classical QAs. These QAs will be presented in Chap. 5.

4.1 Genetic Programming and Evolutionary Algorithms

One can consider GP to be a search method that seeks an optimal program in the search space spanned by computer programs. Each program can be rated by a *fitness*

value that reflects the program's adequacy to solve a problem. If it is also possible to define a neighborhood in the space spanned by the programs then one obtains a *fitness landscape* (see Fig 4.1) by assigning a fitness value to each program.

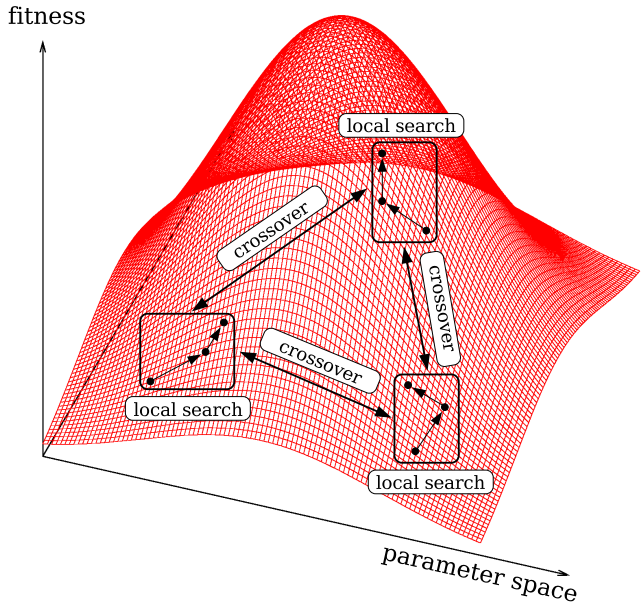

Figure 4.1: This schematic fitness landscape illustrates the main idea of evolutionary algorithms: A population of local search processes scans the fitness landscape. Information gained by these processes is shared by crossover in order to create new local search processes.

Fitness landscapes provide a helpful visualization to compare GP with other search methods:
An *exhaustive search* tests each point of the fitness landscape. Such a method becomes impractical for huge search spaces where a method called *hill climbing* is more appropriate [SR95]. Here some neighboring points of the actual one are tested. The point with the better fitness value becomes the actual one. This method is only successful for landscapes without local optima. To circumvent the possibility of getting stuck in such a local optima it is necessary to also allow a neighboring point of lower fitness to become the actual one. This is provided by the *Metropolis algorithm* [MRR+53]. If the frequency of downhill steps is altered during the search process then one speaks of *simulated annealing* [Kir83].

Apart from GP all search methods presented above visit only one point at each time step. Therefore one runs such algorithms several times in order to gain confidence that a solution found indeed is a good solution. Unfortunately, such iterative runs do not share any topological information of the fitness landscape acquired in former runs. This drawback is tackled by population based search algorithms. They provide mechanisms to share informations gathered by the individuals. If the mechanism of information sharing mimics the biological process responsible for the evolution of

species then one speaks of *evolutionary algorithms*.

Among the most well-known evolutionary algorithms are *genetic algorithms* [Hol73], *evolution strategies* [Rec73, BS02] and GP [Koz92, BNKF98, LP02]. The former two methods are parameter optimization techniques. Genetic algorithms represent these parameters by binary strings, evolution strategies by real-valued vectors.

The search methods we compared GP with are mainly used to solve optimization problems. GP is a modeling system that improves programs automatically through experience. In order to clarify the conceptual differences between optimization and modeling problems we adopt the system analyst's perspective used in the book of Eiben and Smith [ES03]:

A system consists of three components: the inputs, the outputs and an internal model that maps inputs to outputs. If one seeks the inputs for a system whose internal model and outputs are known then one speaks of optimization problems. One famous example is the "traveling salesman problem": One optimizes the sequence of towns that are to be visited in order to minimize the length of the tour across these towns. If, on the other hand, only the inputs and the outputs are known then one speaks of a modeling problem: One is interested in the internal model of the system. This is exactly what GP is used for in this thesis: Consider for example the development of the parity algorithm described in Sec. 5.3.2. The known inputs are the blackboxes X, the outputs are their parities $f(X)$. We are interested in the QC that calculates $f(X)$ from the input X.

4.2 Setting up the GP System

Now we will present the GP system used by us to develop the QAs that will be discussed Chap. 5.

In Sec. 4.2.1 we will describe how QCs are represented by our GP system. Before we will introduce our fitness functions in Sec. 4.4 we will make some preliminary notes about mutation and local search in the context of quantum computation in Sec. 4.2.2. Then we will present our oracle gates in Sec. 4.3. Because we evolved QCs for single issue quantum computers as well as for ensemble quantum computers the section about fitness functions is divided into two parts. A detailed example of an evolutionary run of our GP system can be found in Sec. C.

4.2.1 Representation of Quantum Decision Trees in the GP System

A computation of a blackbox's property can be visualized by a decision tree. According to Buhrman *et al.* a QDT is defined by a sequence of oracle gates $\boldsymbol{O}$ that represent blackbox calls, alternating with unitary transformations $\boldsymbol{U}_i$ [BdW02]:

$$\boldsymbol{A} = \boldsymbol{U}_T \cdot \boldsymbol{O} \cdot \boldsymbol{U}_{T-1} \cdot \boldsymbol{O} \ldots \boldsymbol{U}_2 \cdot \boldsymbol{O} \cdot \boldsymbol{U}_1.$$

The sequence denoted by the unitary transformation $\boldsymbol{A}$ is applied to the initial state $|\psi_{\text{init}}\rangle = |\underline{0}\rangle$. Measuring the readout-qubit of the final state $|\psi_{\text{final}}\rangle = \boldsymbol{A}|\psi_{\text{init}}\rangle$ returns a binary value. If this binary value equals $f(X)$ for all blackboxes then one says that the QDT computes the property $f(X)$ of the blackbox encoded by the oracle gate $\boldsymbol{O}$.

It is adequate to represent QDTs by QCs. In order to do that one needs a complete set of one-qubit and two-qubit gates. Additionally, one needs the set of oracle gates

$\boldsymbol{O}$ representing the blackboxes X. The measurement of a single read-out qubit is sufficient to decide the property $f(X)$ of a blackbox X.

Nevertheless, algorithms like the DJ algorithm answer the problem by a measurement of all qubits (see Sec. 3.4.1). This procedure makes it possible to dispense with the additional read-out qubit and additional quantum operations necessary to encode the answer into this read-out qubit. Now the answer to the posed problem cannot be obtained any more by measuring the state of a single read-out qubit.

The QA has to return different measurement results for those blackboxes X that differ in their property $f(X)$. We decided to perform all possible measurements on the final state $|\psi_{\text{final}}\rangle$ in order to check if one of these measurements returns different results for blackboxes X that differ in their property $f(X)$. If the state $|\psi_{\text{init}}\rangle$ is encoded by n qubits then we perform all n single-qubit projective measurements, all $n(n-1)$ two-qubit projective measurements and so on.[1] Here we only consider projective measurements that project onto computational basis states.

Analogous to Spector *et al.* we have chosen a linear genome to represent QCs in our GP system (see Fig. 4.2) [SBBS99].

A quantum gate is specified by several parameters like rotation angles, control-qubits, etc. Therefore, one has to decide where these additional data are to be stored and how they are to be manipulated by the evolutionary process. The most natural method is to consider a quantum gate and these additional parameters as a unit. The evolutionary process only modifies the unit as a whole.

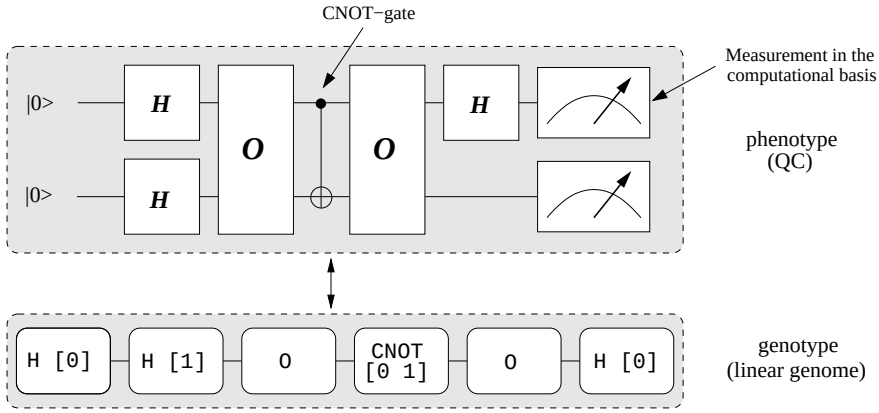

Figure 4.2: In our GP system a QC is represented by a linear list (linear genome) of the quantum gates used by the algorithm. The sequence of the quantum gates is obtained by reading this linear list from the left to the right. `H [1]` denotes a Hadamard gate $\boldsymbol{H}$ that is applied to qubit 1. `CNOT [0 1]` denotes a $\boldsymbol{CNOT}$ gate with qubit 0 the control and qubit 1 the target qubit.

[1] This corresponds to a total of 2^n measurements. This is done to evaluate the QC and therefore has no influence on the scalability of the QC.

4.2.1.1 Genetic Operators

Our GP system uses a generational approach: After each time step a new population of individuals is generated from the old one. To generate new individuals one uses so called *genetic operators* - i.e., *mutations*, *recombination* and *reproduction*. To achieve this one adopts mechanisms that mirror the mutation and recombination processes assumed to be the main ingredients in the evolutionary development of species in nature. The concepts of molecular biology that describe recombination and mutation processes on the molecular level suggest how mutation and recombination can be incorporated into the GP system:

- A mutation alters a functional unit of the linear genome - i.e., a quantum gate has to be deleted, inserted or exchanged with another quantum gate (see Fig. 4.3).
- The recombination is realized by a crossover process. In each of the two parent individuals a sublist is chosen at random. These two sublists are exchanged between the parents (see Fig. 4.4).
- We have chosen to copy the best individual of each generation unaltered to the new generation. This process is called reproduction and the main reason in doing so is not to loose the best solution found so far.

In each generation we generated some individuals at random in order to compare the efficiency of the evolutionary process with the efficiency of an exhaustive search.

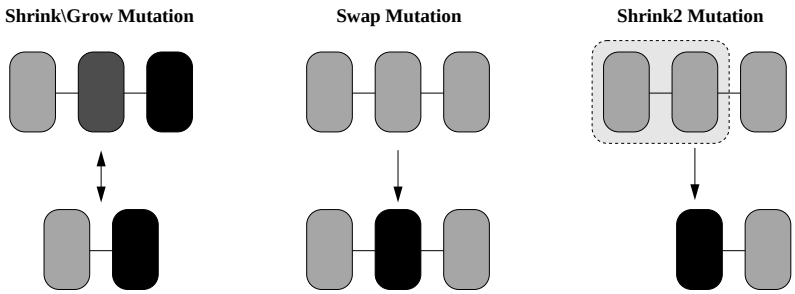

Figure 4.3: This figure shows the four different mutations used by our GP system. The *shrink mutation* and the *grow mutation* are shown in the leftmost graphic: A shrink mutation chooses a quantum gate at random and deletes it. A grow mutation inserts a quantum gate at a random position in the genome. The next graphic shows a *swap mutation*: A quantum gate is replaced by another quantum gate. The rightmost graphic shows an additional mutation process that makes it possible to concatenate two, not necessarily neighboring, quantum gates.

4.2.1.2 Tournament Selection

The individuals that are to be altered by genetic operators were chosen by a tournament selection: Dependent of a mutation or crossover to occur the best or the two best individuals are selected from a subset of all individuals. The size of this subset

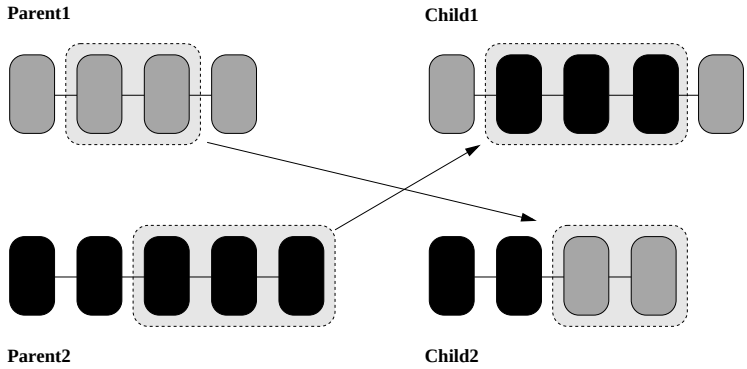

Figure 4.4: This figure shows a recombination process: A connected sublist in each individual is chosen at random. Then these two sublists are exchanged between the individuals. The different gray scales are used to indicate the origin of the quantum gates.

influences the selection pressure - i.e., the bigger the subset the larger the selection pressure.

4.2.2 Genetic Operators and Local Search

In the beginning of this chapter we introduced the concept of fitness landscapes to compare GP with other search methods. The main problem in introducing this concept is that it is not obvious how the neighborhood of a GP individual should be defined.

One possibility is to define those individuals to be neighbors that can be transformed into each other by a single application of a genetic operator. This definition is useful when no further information about the structure of the problem is known. If one evolves QCs then such a definition of the neighborhood is misleading: It does not respect the metric of the Hilbert space of QCs.

In the next two paragraphs we will show how a norm and thus a distance measure can be defined on the set of QCs. We will use this measure to calculate the step-length of the genetic operators used by our GP system. Finally we will show that this distance measure provides an upper bound for the maximal difference between the measurement results of two QCs.

4.2.2.1 Distance Measure for Genetic Operators

Any quantum computation can be represented by unitary transformations (see Sec. 2.7). The set of unitary operators forms a vector space. An inner product between two unitary operators $\boldsymbol{U}$ and $\boldsymbol{V}$ can be defined by [NC00]:

$$\langle \boldsymbol{U}, \boldsymbol{V} \rangle \equiv \mathrm{tr}\{\boldsymbol{U}^{+}\boldsymbol{V}\}. \tag{4.1}$$

Now it is possible to define the norm $\|\boldsymbol{U}\|$ of the unitary operator $\boldsymbol{U}$:

$$\|\boldsymbol{U}\| \equiv \sqrt{\langle \boldsymbol{U}, \boldsymbol{U} \rangle}. \tag{4.2}$$

The norm makes it possible to introduce a step-length of genetic operators: Let $\boldsymbol{U}$ denote a QC that is transformed by a grow mutation to another QC denoted by $\boldsymbol{V}$. This grow mutation inserts the additional quantum gate $\boldsymbol{A}$. It is sufficient to consider three cases:

1. The genetic operator maps $\boldsymbol{U}$ to $\boldsymbol{V} = \boldsymbol{A}\boldsymbol{U}$.
2. The genetic operator maps $\boldsymbol{U}$ to $\boldsymbol{V} = \boldsymbol{U}\boldsymbol{A}$.
3. The genetic operator maps $\boldsymbol{U} = \boldsymbol{U}_1\boldsymbol{U}_2$ to $\boldsymbol{V} = \boldsymbol{U}_1\boldsymbol{A}\boldsymbol{U}_2$.

The calculations are the same for all three cases, hence we only consider the last one. At first we define the distance $d(\boldsymbol{U}, \boldsymbol{V})$ between two unitary operators $\boldsymbol{U}$ and $\boldsymbol{V}$ by:

$$d(\boldsymbol{U}, \boldsymbol{V}) = \|\boldsymbol{U} - \boldsymbol{V}\|.$$

We get:

$$\begin{aligned} d(\boldsymbol{U}, \boldsymbol{V}) &= \|\boldsymbol{U}_1\boldsymbol{U}_2 - \boldsymbol{U}_1\boldsymbol{A}\boldsymbol{U}_2\| = \sqrt{\mathrm{tr}\{2 \cdot \mathbf{1} - \boldsymbol{U}_2^+\boldsymbol{A}\boldsymbol{U}_2 - \boldsymbol{U}_2^+\boldsymbol{A}^+\boldsymbol{U}_2\}} \\ &= \sqrt{\mathrm{tr}\{2 \cdot \mathbf{1} - \boldsymbol{A} - \boldsymbol{A}^+\}}. \end{aligned}$$

The last equation follows due to the invariance of the trace under cyclic permutations of its arguments ($\mathrm{tr}\{\boldsymbol{A}\boldsymbol{U}\} = \mathrm{tr}\{\boldsymbol{U}\boldsymbol{A}\}$).

Now we can calculate the step-length of the genetic operators:

$$\begin{array}{lll} \left.\begin{array}{l} \boldsymbol{A} = \boldsymbol{R}_x(\alpha) = \exp\left(-\imath\frac{\alpha}{2}\boldsymbol{\sigma}_x\right) \\ \boldsymbol{A} = \boldsymbol{R}_y(\alpha) = \exp\left(-\imath\frac{\alpha}{2}\boldsymbol{\sigma}_y\right) \end{array}\right\} & \Rightarrow & d(\boldsymbol{U}, \boldsymbol{V}) \approx \frac{\alpha}{2} \cdot \|\mathbf{1}\| \text{ if } \alpha \ll 1, \\ \boldsymbol{A} = \boldsymbol{H} & \Rightarrow & d(\boldsymbol{U}, \boldsymbol{V}) = \sqrt{2} \cdot \|\mathbf{1}\|, \\ \boldsymbol{A} = \boldsymbol{CNOT} & \Rightarrow & d(\boldsymbol{U}, \boldsymbol{V}) = 1 \cdot \|\mathbf{1}\|. \end{array}$$

Here we only considered grow mutations that insert an additional quantum gate $\boldsymbol{A}$.

The calculations are also valid for shrink mutations as justified by the following argument: Let the sequence $\boldsymbol{U}_1 \cdot \boldsymbol{A} \cdot \boldsymbol{U}_2$ of unitary operators denote a QC. A shrink mutation that deletes $\boldsymbol{A}$ is equivalent to a grow mutation that inserts the inverse $\boldsymbol{A}^{-1}$ next to $\boldsymbol{A}$.

A swap mutation or crossover can be described by a combined shrink and grow mutation, hence it is covered by our calculations.

The distance $d(\boldsymbol{U}, \boldsymbol{V})$ of two unitary operators $\boldsymbol{U}$ and $\boldsymbol{V}$ provides an upper bound for the difference $1/2 \cdot |\langle \boldsymbol{M} \rangle_{\boldsymbol{U}} - \langle \boldsymbol{M} \rangle_{\boldsymbol{V}}| / \|\boldsymbol{M}\|$ between the expectation value $\langle \boldsymbol{M} \rangle_{\boldsymbol{V}}$ and $\langle \boldsymbol{M} \rangle_{\boldsymbol{U}}$. We introduced the notation $\langle \boldsymbol{M} \rangle_{\boldsymbol{U}}$ to abbreviate the term $\langle \psi \boldsymbol{U}^+ | \mathbf{M} | \boldsymbol{U}\psi \rangle$. Now we give a proof of this upper bound which, in its main steps, resembles the proof given in [NC00] for a similar inequality:

Proof:

We start by noting that the QC described by the unitary operation $\boldsymbol{U}$ maps the initial state $\boldsymbol{\varrho}$ to $\boldsymbol{U}\boldsymbol{\varrho}\boldsymbol{U}^+$, that described by $\boldsymbol{V}$ maps $\boldsymbol{\varrho}$ to $\boldsymbol{V}\boldsymbol{\varrho}\boldsymbol{V}^+$. The norm $\|\boldsymbol{M}\|$ of the Hermitian operator $\boldsymbol{M}$ can be defined analogous to Eq. (4.2).

Because the expectation value $\langle \boldsymbol{M} \rangle_{\boldsymbol{U}}$ is calculated via $\langle \boldsymbol{M} \rangle_{\boldsymbol{U}} = tr\{\boldsymbol{U}\boldsymbol{\varrho}\boldsymbol{U}^+\}$ the difference $|\langle \boldsymbol{M} \rangle_{\boldsymbol{U}} - \langle \boldsymbol{M} \rangle_{\boldsymbol{V}}|$ between the measurement results is given by:

$$\begin{aligned} |\langle \boldsymbol{M} \rangle_{\boldsymbol{U}} - \langle \boldsymbol{M} \rangle_{\boldsymbol{V}}| &= |\mathrm{tr}\{\boldsymbol{U}^+\boldsymbol{M}\boldsymbol{U}\boldsymbol{\varrho}\} - \mathrm{tr}\{\boldsymbol{V}^+\boldsymbol{M}\boldsymbol{V}\boldsymbol{\varrho}\}| \\ &= |\mathrm{tr}\{\boldsymbol{U}^+\boldsymbol{M}\left(\boldsymbol{U}-\boldsymbol{V}\right)\boldsymbol{\varrho}\} + \mathrm{tr}\{\left(\boldsymbol{U}-\boldsymbol{V}\right)^+\boldsymbol{M}\boldsymbol{V}\boldsymbol{\varrho}\}| \\ &\leq |\mathrm{tr}\{\boldsymbol{\varrho}\boldsymbol{U}^+\boldsymbol{M}\left(\boldsymbol{U}-\boldsymbol{V}\right)\}| + |\mathrm{tr}\{\left(\boldsymbol{U}-\boldsymbol{V}\right)^+\boldsymbol{M}\boldsymbol{V}\boldsymbol{\varrho}\}| \\ &= |\langle \boldsymbol{M}^+\boldsymbol{U}\boldsymbol{\varrho}^+, \left(\boldsymbol{U}-\boldsymbol{V}\right)\rangle| + |\langle \left(\boldsymbol{U}-\boldsymbol{V}\right), \boldsymbol{M}\boldsymbol{V}\boldsymbol{\varrho}\rangle| \\ &\leq \|\boldsymbol{M}\boldsymbol{U}^+\boldsymbol{\varrho}\| \cdot \|\boldsymbol{U}-\boldsymbol{V}\| + \|\boldsymbol{U}-\boldsymbol{V}\| \cdot \|\boldsymbol{M}\boldsymbol{V}\boldsymbol{\varrho}\| \end{aligned}$$

Here we used the definition of the inner product $< \boldsymbol{A}, \boldsymbol{B} >$ of two operators $\boldsymbol{A}$ and $\boldsymbol{B}$ stated in Eq. (4.1). The last line follows from Cauchy-Schwarz's inequality. Before we proceed we show that the norm $\|\boldsymbol{M}\boldsymbol{U}^+\boldsymbol{\varrho}\|$ does no depend on the unitary operator $\boldsymbol{U}^+$:

$$\begin{aligned} \|\boldsymbol{M}\boldsymbol{U}^+\boldsymbol{\varrho}\| &= \sqrt{tr\left\{\left(\boldsymbol{M}\boldsymbol{U}^+\boldsymbol{\varrho}\right)^+\left(\boldsymbol{M}\boldsymbol{U}^+\boldsymbol{\varrho}\right)\right\}} = \sqrt{tr\{\boldsymbol{M}^2\left(\boldsymbol{U}^+\boldsymbol{\varrho}^2\boldsymbol{U}\right)\}} \\ &= \sqrt{tr\{\boldsymbol{M}^2\left(\boldsymbol{U}^+\boldsymbol{\varrho}\boldsymbol{U}\right)^2\}} = \sqrt{\langle \boldsymbol{M}^2, \left(\boldsymbol{U}^+\boldsymbol{\varrho}\boldsymbol{U}\right)^2\rangle} \\ &\leq \sqrt{\|\boldsymbol{M}^2\| \cdot \|\left(\boldsymbol{U}^+\boldsymbol{\varrho}\boldsymbol{U}\right)^2\|} \leq \|\boldsymbol{M}\| \cdot \|\boldsymbol{\varrho}\| \end{aligned}$$

The last inequality follows because of $\|\boldsymbol{\varrho}^2\| \leq \|\boldsymbol{\varrho}\|^2$. This inequality holds for any Hermitian operator $\boldsymbol{\varrho}$:

$$\|\boldsymbol{\varrho}^2\| = \sqrt{tr\{\boldsymbol{\varrho}^4\}} = \sqrt{\sum_i (\boldsymbol{\varrho})_{ii}^4} \leq \sqrt{\left(\sum_i (\boldsymbol{\varrho})_{ii}^2\right)^4} = \|\boldsymbol{\varrho}\|^2$$

According to these results one gets:

$$|\langle \boldsymbol{M} \rangle_{\boldsymbol{U}} - \langle \boldsymbol{M} \rangle_{\boldsymbol{V}}| \leq 2 \cdot \|\boldsymbol{U}-\boldsymbol{V}\| \cdot \|\boldsymbol{M}\| \cdot \|\boldsymbol{\varrho}\| \leq 2 \cdot \|\boldsymbol{U}-\boldsymbol{V}\| \cdot \|\boldsymbol{M}\|.$$

The last line follows because of $\|\boldsymbol{\varrho}\| \leq 1$ (see Sec. 2.7). □

According to the last result the distance between two QCs is a measure for the maximal difference in the corresponding measurement results. The fitness function is calculated using these measurement results. It is desirable to design a fitness function that is continuous in the step-length of the genetic operators. Such a fitness function would allow for local searches. The design of a fitness function that partly fulfills this requirements is described in Sec. 4.4.

It is possible to allow for local search if the set of mutation operators is chosen carefully. In our case this is done by adjusting the parameter α. The operators $\boldsymbol{R}_x(\alpha)$, $\boldsymbol{R}_y(\alpha)$ and $\boldsymbol{CNOT}$ form a complete set. Therefore, it is possible to evolve each QC by only small mutations.[2] A search that is only based on small mutations is inefficient, therefore we used a discrete set of parameters $\alpha \in [-\pi/2, \pi/2[$ with sufficiently small subdivisions.

[2] The $\boldsymbol{CNOT}$ gate does not depend on a continuous parameter. The step length of a mutation that inserts a $\boldsymbol{CNOT}$ gate is fixed and cannot be adjusted. Therefore, we sometimes used the adjustable two-qubit gate $\boldsymbol{R}_{zz}(\alpha) = \exp\left(-\imath\frac{\alpha}{4}\sigma_z \otimes \sigma_z\right)$. Together with the one-qubit gates $\boldsymbol{R}_x(\alpha)$ and $\boldsymbol{R}_y(\alpha)$ these gates provide a complete set of unitary operations [KG00].

4.3 Oracle Gates

In order to see if a QC correctly calculates the property $f(X)$ of the blackbox X one has to check this circuit for all blackboxes. The blackboxes are encoded into a quantum system via the oracle gates $\boldsymbol{O}$. Thus, these gates provide the fitness cases the QC is to be tested with. It is possible that the number of blackboxes that are to be tested grows super-exponentially: The DJ problem presented in Sec. 3.4.1 provides an illustrative example: For a one-qubit query register one has to test 2 constant blackboxes $X_1 = (0,0)$, $X_2 = (1,1)$ with $f(X_{1,2}) = 0$ and 2 balanced blackboxes $X_3 = (0,1)$, $X_4 = (1,0)$ with $f(X_{3,4}) = 1$. For a two-qubit query register one has to test 2 constant and 6 balanced blackboxes. For an n-qubit query register one has to test 2 constant and $N!/[(N/2)!]^2$ balanced blackboxes, with $N = 2^n$. Using Stirling's formula one gets $N!/[(N/2)!]^2 \sim 2^N/N^{1/2}$ for $N \gg 1$.

This example shows that not only the effort to simulate a quantum system grows exponentially with the number of qubits, but also the number of oracles that are to be tested increases quickly. This renders an investigation of QCs with many qubits impractical. One should therefore strive to reduce the number of oracles to be tested.

We could run the GP system on a subset of all oracles. This approach is hampered by the fact that the proportion of "hard" blackboxes to the number of "not-hard" blackboxes can decrease exponentially in the number of query qubits.[3] Hence, it is likely that a randomly chosen subset does not anymore represent the problem to be solved. This result indicates that in general the investigation of QAs cannot be simplified by running the GP system on a small subset of fitness cases as long as the size of this subset does not increase exponentially in the number of query qubits.

Another possibility is to use an alternative encoding of the blackbox values into a quantum state that enables a reduction in the number of oracles to be tested.

The DJ problem for a single query qubit can be used to illustrate this procedure. Here the constant blackboxes X_1, X_2 with $f(X_{1,2}) = 0$ are to be distinguished from the balanced blackboxes X_3, X_4 with $f(X_{3,4}) = 1$.

If one employs the usual definition of the oracle gate [NC00]:

$$\boldsymbol{O}|i\rangle|0\rangle = |i\rangle|x_i\rangle, \quad i \in \mathbb{N}, x_i \in \{0,1\}, \tag{4.3}$$

one has to test each of the four different oracles. Because the DJ problem can also be solved using the following oracle gate [CKH98]:

$$\boldsymbol{O}|i\rangle = (-1)^{x_i}|i\rangle, \tag{4.4}$$

it is possible to find a QC that solves this problem by only testing two oracles.

With the latter definition we get the same oracle, up to a global phase shift, for the two constant blackboxes: $\boldsymbol{O}_{1,2} = \pm\mathbf{1}$. For the two balanced blackboxes one gets:

[3] The term "hard" denotes blackboxes whose property can only be calculated by a number of oracle calls that grows exponentially in the number of query qubits. Ambainis *et al.* [AdW01] presented a blackbox problem whose average-case query complexity on a quantum computer is exponentially smaller (in the number of qubits) than the average-case query complexity on a classical computer. The worst-case query complexity of a QA that computes a total Boolean function is polynomially related to the worst-case query complexity of a classical algorithm that computes such a function (see Sec. 3.2.2). This result applies to the problem investigated by Ambainis *et al.* Therefore, the worst-case query complexity of quantum computers can be exponentially higher than their average-case query complexity!

$\boldsymbol{O}_{3,4} = \pm\boldsymbol{\sigma}_x$. Because global phase shifts are not measurable it is not necessary to test all 4 different oracles but only the oracles $\boldsymbol{O}_1 = \mathbf{1}$ and $\boldsymbol{O}_3 = \boldsymbol{\sigma}_x$. A further advantage of this method is that one needs no additional output register. This would be the case for the oracle definition in Eq. (4.3).

It depends on the property $f(X)$ one is interested in whether an encoding of the blackbox entry x_i into the phase $(-1)^{x_i}$ is possible. If, for instance, one wants to calculate the property $f(X_1) = 0$ and $f(X_2) = 1$ with $X_1 = (0,0)$ and $X_2 = (1,1)$ then it is impossible to encode the blackbox entries into phase shifts. Both cases become indistinguishable for a quantum system because they only differ by a global phase shift. To see whether an encoding into a local phase shift via Eq. (4.4) is possible one has to check if for every pair of blackboxes $X = (x_0, x_1, \ldots, x_n)$ and $\bar{X} = (\bar{x}_0, \bar{x}_1, \ldots, \bar{x}_n)$ the condition $f(X) = f(\bar{X})$ holds. Here $\bar{x}_k$ denotes the negation of the binary variable x_k.

Up to now we only considered blackboxes X whose entries were binary values. In Simon's problem [Sim94] one investigates blackboxes $X = (x_0, x_1, ..., x_{N-1})$ with $x_i \in \{0, 1, ..., l-1\}$ and $l \in \mathbb{N}$. If one applies the approach of Eq. (4.3) then one needs an output register size of $\log_2(l)$ qubits. Both, in simulation and experiment, only a few qubits are realizable up to now. Therefore, this approach is not feasible. One possibility to reduce the number of qubits is to encode the values $x_i \in \{0, 1, ..., l-1\}$ into complex phase shifts:

$$\boldsymbol{O}|i\rangle = (\xi)^{x_i}|i\rangle, \quad \xi = e^{2\pi\imath/l}. \tag{4.5}$$

One also can use a combined approach where the blackbox entries $x_i \in \{0, 1, ..., l\}$ with $l \in \mathbb{N}$ are encoded partially into a phase shift and partially into an output register: In this case we represent the blackbox entry x_i by the decomposition $x_i^{(\text{phase})}\underline{x_i}^{(\text{xor})}$. The part $x_i^{(\text{phase})}$ is to be encoded into a phase shift. The part $\underline{x_i}^{(\text{xor})}$ is to be encoded into an output register.

Consider, for example, blackbox entries $x_i \in \{0, \ldots, 15\}$. With two output qubits one can only encode 4 different values. Therefore, one needs another 4 different phase shifts to unambiguously encode x_i. The value $x_i = 13$ would be decomposed into $x_i^{(\text{phase})} = 3$ and $\underline{x_i}^{(\text{xor})} = 01$. This decomposition follows from the binary representation of 13: 1101. The two rightmost bits are the assignment for $\underline{x_i}^{(\text{xor})}$, the integer represented by the two leftmost qubits is the value of the phase shift. In general the oracle gate is defined by:

$$\boldsymbol{O}|\underline{i}\rangle|\underline{b}\rangle = (\xi)^{x_i^{(\text{phase})}}|\underline{i}\rangle|\underline{b} \oplus \underline{x_i}^{(\text{xor})}\rangle, \tag{4.6}$$

with $i \in \{0, 1, ..., N-1\}$ and $\underline{b}, \underline{x_i} \in \{0,1\}^m$.

In Sec. 5.2 we will present a QA that was evolved using such a kind of oracle gates.

4.4 Fitness Functions

The definition of the fitness function is the most sensitive part in setting up the GP system. Here one has to make assumptions that might turn out to be ill-suited for evolving QCs. As shown in Sec. 4.2.2.1 the mathematical structure of quantum physics allows to define a metric on the space of QCs. This enables us to assign a

step-length to genetic operators. Small mutations, according to this measure, lead to small changes in the expectation values. The fitness function depends on these expectation values, hence it is desirable that it is continuous in the step-length of the genetic operators. All of our QCs were evolved using a fitness function that did not fully obey this condition.

4.4.1 Fitness Function for Single Issue Quantum Computers

The problems treated by our GP system were the parity problem and a special case of the hidden subgroup problem. Nevertheless, we will motivate the fitness function of our GP system by means of the DJ problem (see Sec. 3.4.1).

The DJ problem for 4 blackbox elements is to distinguish between the constant blackboxes $X_1 = (0,0,0,0)$, $X_2 = (1,1,1,1)$ with $f(X_{1,2}) = 0$ and the balanced blackboxes:

$$\begin{array}{lll} X_3 = (0,0,1,1), & X_4 = (1,1,0,0), & X_5 = (0,1,0,1), \\ X_6 = (1,0,1,0), & X_7 = (0,1,1,0), & X_8 = (1,0,0,1), \end{array}$$

with $f(X_{3,4,5,6,7,8}) = 1$. According to Collins *et al.* this problem can be solved by encoding the blackbox entries into phase shifts via the oracle gates defined by Eq. (4.4) [CKH98]. Thus, fitness depends on the measurement results for the 4 blackboxes $X_1 = (0,0,0,0)$, $X_3 = (0,0,1,1)$, $X_5 = (0,1,0,1)$ and $X_7 = (0,1,1,0)$. The oracle gates of the remaining blackboxes differ from the oracle gates $\boldsymbol{O}_1$, $\boldsymbol{O}_3$, $\boldsymbol{O}_5$ and $\boldsymbol{O}_7$ that represent the four blackboxes mentioned above by a global phase shift (see also Sec. 4.3).

Tab. 4.1 shows measurement probabilities for a QC applied to the input state $|\psi_{\text{init}}\rangle = |00\rangle$. How well does this circuit decide between constant and balanced blackboxes?

	$prob(\|00\rangle)$	$prob(\|01\rangle)$	$prob(\|10\rangle)$	$prob(\|11\rangle)$
$X = (0,0,0,0)$	0.8	0.05	0.05	0.1
$X = (0,0,1,1)$	0.15	0.25	0.25	0.35
$X = (0,1,0,1)$	0.1	0.15	0.15	0.6
$X = (0,1,1,0)$	0.1	0.5	0.3	0.1

Table 4.1: Measurement probabilities for a hypothetical QC. $prob(|i_1 i_0\rangle)$ denotes the probability of measuring quantum state $|i_1 i_0\rangle$ after applying the QC to $|00\rangle$.

We consider a quantum state $|i_1 i_0\rangle$ to be measurable with a sufficient probability if $prob(|i_1 i_0\rangle) > 1/N$. Here $N = 2^n$ is the number of different quantum states for n qubits and $prob(|i_1 i_0\rangle)$ the probability to measure state $|i_1 i_0\rangle$. A binary variable $b_{i_1 i_0}$ indicates whether the quantum state $|i_1 i_0\rangle$ fulfills this condition (see Tab. 4.2).

One has to check if there are indices i_1 and i_0 so that $b_{i_1 i_0}$ is equal to 1 for constant as well as balanced blackboxes. In this case the QC is not able to distinguish between both types of blackboxes. Otherwise, we consider the circuit a promising candidate for solving the problem.

According to Tab. 4.2 only the constant blackbox is mapped to state $|\psi_{\text{final}}\rangle = |00\rangle$ with a probability greater than 1/4. Therefore, the QC is able to distinguish

	b_{00}	b_{01}	b_{10}	b_{11}
$X = (0,0,0,0)$	1	0	0	0
$X = (0,0,1,1)$	0	0	0	1
$X = (0,1,0,1)$	0	0	0	1
$X = (0,1,1,0)$	0	1	1	0

Table 4.2: This table is obtained from Tab. 4.1 by setting the binary value $b_{i_1 i_0} = 1$ if $prob(|i_1 i_0\rangle) > 1/4$, otherwise $b_{i_1 i_0} = 0$.

between constant and balanced blackboxes. However, Tab. 4.1 shows that sometimes a balanced blackbox is also classified as being constant. It is better, therefore, to have the fitness function depend on at least two parameters.

The first parameter of the fitness function is called `clash` and quantifies how often it is not possible to distinguish between constant and balanced blackboxes. Another parameter, `worst_error`, denotes the highest probability of a misclassification. In the example above we have `worst_error` $= 0.2$, the error probability that a constant function is classified to be balanced. Further parameters are `avg_error` to denote the average error, `oracles` to denote the number of oracle gates and `length` to denote the total number of quantum gates.

Similar to the approach used by Spector *et al.* these parameters were used to create a lexicographic fitness function represented by a vector of the form [SBBS99, Spe04]:

$$f = \begin{pmatrix} \texttt{clash} \\ \texttt{worst\_error} \\ \texttt{avg\_error} \\ \texttt{oracles} \\ \texttt{length} \end{pmatrix}.$$

The position of the parameters in this vector represents their priority, decreasing from top to bottom.

Unfortunately, the component `clash` is not continuous in the step-length of the genetic operators. When `clash` is equal to 0 the parameter `worst_error` becomes significant which is continuous in the step-length of the genetic operators.

Our fitness function is to be contrasted with that of Spector *et al.* who measure the state of a single qubit [SBBS99, Spe04]. The advantage of our approach is that an evolved QC that solves the DJ problem has the possibility to resemble the original one. Therefore, we do not need additional quantum gates necessary to solve the problem by the measurement of a single read-out qubit. Such additional quantum gates would make it difficult to extract the functionality and thus the scalability of a QC.

4.4.2 Fitness Function for Ensemble Quantum Computers

An NMR-QC realizes qubits by the spin states of the molecule's spin-1/2 nuclei. Such a quantum computer performs its computations on an ensemble of molecules, hence the fitness function has to be derived from the expectation values $\langle \boldsymbol{I}_x^{(i)} \rangle$ of the measurement operator $\boldsymbol{I}_x^{(i)}$.

Consider for example a two-qubit system. According to Sec. 2.8.2 the initial state is described by:

$$\boldsymbol{\varrho}_{th} \sim \frac{1}{4}\left(\omega_0 \boldsymbol{I}_z^{(0)} + \omega_1 \boldsymbol{I}_z^{(1)}\right). \tag{4.7}$$

In contrast to Eq. (2.25) we skipped the term proportional to the identity matrix as it does not contribute to the expectation values $\langle \boldsymbol{I}_x^{(i)} \rangle$.

The measurement of the i-th spin's magnetization along the x-axis is calculated by:

$$\langle \boldsymbol{I}_x^{(i)} \rangle = \mathrm{tr}\{\boldsymbol{I}_x^{(i)} \boldsymbol{\varrho}\}. \tag{4.8}$$

All combinations of measuring the spins' magnetization are performed in order to check if one of these returns different results for blackboxes X that differ in their property $f(X)$. In our example one calculates $\langle \boldsymbol{M} \rangle$ for $\boldsymbol{M} = \boldsymbol{I}_x^{(0)}$, $\boldsymbol{M} = \boldsymbol{I}_x^{(1)}$ and $\boldsymbol{M} = \boldsymbol{I}_x^{(0)} + \boldsymbol{I}_x^{(1)}$.

We divided the interval $[-|\langle \boldsymbol{M} \rangle|, \ldots, |\langle \boldsymbol{M} \rangle|]$ of possible measurement results into several disjoint sub-intervals.[4] This makes it possible to quantify the success of a promising circuit: If the measurement results for both types of blackboxes belong to different sub-intervals then we consider the circuit a promising candidate for solving the problem.

To decide how well this circuit can distinguish between both types of blackboxes we calculated the minimal distance between the corresponding measurement results. The variable `min_dist` denotes this minimal distance whereas the variable `avg_dist` denotes the average distance.

The fitness function is represented by the vector:

$$f = \begin{pmatrix} \texttt{clash} \\ -\texttt{min\_dist} \\ -\texttt{avg\_dist} \\ \texttt{oracles} \\ \texttt{length} \end{pmatrix}.$$

We used a lexicographic ordering of the vector components with `clash` the most significant one. An optimal algorithm would have minimal values in all components of its fitness function.[5]

Unfortunately, the component `clash` is not continuous in the step-length of the genetic operators. When `clash` is equal to 0 the parameter `min_distance` becomes significant. This parameter is continuous in the step-length of the genetic operators.

[4] With the operator norm $||\boldsymbol{M}|| = \sqrt{\mathrm{tr}\{\boldsymbol{M}^+ \boldsymbol{M}\}}$ and Cauchy-Schwarz's inequality one gets $|\langle \boldsymbol{M} \rangle| \leq \|\boldsymbol{M}\| \cdot \|\boldsymbol{\varrho}_{th}\|$. Thus, the interval of possible measurement results can easily be calculated.

[5] The variables `min_distance` and `avg_distance` are to be maximized. Therefore, we used their negative values in the definition of the fitness function to remain consistent with the condition that an optimal fitness function has minimal values in all of its components.

5 Evolved Quantum Algorithms

In this chapter we will present QAs that were developed with the help of the GP system introduced in the last chapter.

At first we will discuss a QA that solves a modification of the DJ problem. A literature research revealed that this algorithm was already proposed by Chi *et al.* [CKL01]. Nevertheless, we will present this algorithm because it serves as a proof of principle that oracle gates which encode blackbox entries into complex phase shifts can successfully be implemented by our GP system.

The problem we will present in Sec. 5.2 is another modification of the DJ problem. The GP system returned a probabilistic quantum circuit that was slightly better than any classical circuit. Unfortunately, the scaling properties were not identifiable.

With the hybrid oracle gates defined in Eq. (4.6) the GP system returned exact better-than-classical quantum circuits. It was possible to discover the problem's structure and the QA to solve it. In Sec. 5.2 we will demonstrate that this problem is a special case of the hidden subgroup problem. Our QA solves this special case exactly and needs less oracle calls than QAs known to us.

Finally, we will present formerly unknown better-than-classical QAs that solve the parity problem in Sec. 5.3. At first we will discuss an algorithm that solves this problem on a single issue quantum computer. It is optimal in terms of additional gate operations required. This algorithm calls the oracle $N/2$ times and thus meets the lower bound of oracle calls established by Beals *et al.* and Farhi *et al.* [FGGS98, BBC$^+$01] (see Sec. 3.2.2).

We also used this problem to test an extended version of our GP system which is able to simulate ensemble quantum computers. These tests returned a QA that solves the parity problem on the thermal state of an NMR-QC. It requires less oracle calls then the single issue QA. Further runs of the GP system for ensemble quantum computers indicated that the number of oracle calls can be reduced further provided the signal-to-noise ratio is sufficiently high. Investigations on this topic revealed that there exists a whole series of QAs for ensemble quantum computers that solve the parity problem with less oracle calls than the known lower bounds. This series of QAs will be presented in Sec. 5.3.4.

In Sec. 5.3.6 we will discuss the strictness of the lower bounds for the parity problem proven by Beals *et al.* [BBC$^+$01].

5.1 A Quantum Algorithm that Solves a Modification of DJ's Problem

We started our investigation with a generalization of DJ's problem. This problem was examined to test oracle gates that encode non-binary blackbox elements $x_i \in \{0, 1, ..., l-1\}$, with $l \in \mathbb{N}$, into phase shifts.

We wanted the GP system to find a QC that distinguishes between blackboxes $X \in A$ with $A = \{(a,a,a,a)\}$ and blackboxes $X \in B$ with $B = \{(a,b,c,d)\}$ for $a \neq b \neq c \neq d$ and $a,b,c,d \in \{0,1,2,3\}$. The set of blackboxes is a genuine subset of all possible blackboxes ($(A \cup B) \subset \{0,1,2,3\}^4$). Therefore, the property $f(X)$, with $f(X) = 0$ when $X \in A$ and $f(X) = 1$ when $X \in B$, is a partial Boolean function. Hence, the lower bounds that were presented in Sec. 3.2.2 do not hold for this problem.

For two query qubits the oracle gate $\boldsymbol{O}$ defined by Eq. (4.5) is of the form:

$$\boldsymbol{O}|i\rangle = \exp\left(\imath\,\frac{\pi}{2}\right)^{x_i}|i\rangle = (\imath)^{x_i}|i\rangle. \tag{5.1}$$

From now on we use $\boldsymbol{O}_X$ to denote the oracle gate that represents the blackbox X:

$$\boldsymbol{O}_{(0,1,2,3)} = \begin{pmatrix} (\imath)^0 & 0 & 0 & 0 \\ 0 & (\imath)^1 & 0 & 0 \\ 0 & 0 & (\imath)^2 & 0 \\ 0 & 0 & 0 & (\imath)^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \imath & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -\imath \end{pmatrix}.$$

All constant blackboxes $X \in A$ differ by a global phase shift:

$$\begin{aligned} \boldsymbol{O}_{(0,0,0,0)} &= \mathbf{1}, & \boldsymbol{O}_{(1,1,1,1)} &= \imath \cdot \mathbf{1}, \\ \boldsymbol{O}_{(2,2,2,2)} &= -1 \cdot \mathbf{1}, & \boldsymbol{O}_{(3,3,3,3)} &= -\imath \cdot \mathbf{1}. \end{aligned}$$

Quantum states with different global phase factors cannot be distinguished. Hence, these blackboxes can be realized by the identity operator $\mathbf{1}$: Only $X = (0,0,0,0)$ has to be implemented. A similar result is obtained for the blackboxes $X \in B$. Here, it is sufficient to only test 6 out of 24 blackboxes:

$$\begin{aligned} &X = (3,2,1,0), \quad X = (2,3,0,1), \quad X = (3,1,2,0), \\ &X = (0,2,1,3), \quad X = (1,0,2,3), \quad X = (0,1,3,2). \end{aligned}$$

The total number of fitness cases can be reduced by a factor of 4 if one encodes the blackbox entries into phase shifts. Thus, the evaluation of a QC by the GP system is reduced by a factor 4×4 in comparison to the usual approach that stores the answer of a blackbox query in an additional two-qubit output register. The second factor 4 is due to the increased costs necessary to simulate a quantum system with two additional qubits. This result indicates that a clever encoding of the blackboxes by oracle gates considerably speeds up the time necessary to evaluate a QC.

The oracle gates that represent blackboxes $X \in B$ can be divided in two classes, those that can be decomposed into a tensor product of two one-qubit gates and those that can't. Consider, for example, the oracle gate that represents the blackbox $X = (3,2,1,0)$:

$$\boldsymbol{O}_{(3,2,1,0)} \equiv \begin{pmatrix} -\imath & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & \imath & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \otimes \begin{pmatrix} c & 0 \\ 0 & d \end{pmatrix},$$

with $a = -b$, $c = \imath d$ and $b, d = 1$. The oracle gate that represents $X = (1,0,2,3)$ cannot be decomposed in such a manner. Hence, the problem investigated here has

entangling oracle gates. The DJ problem for two query qubits can be solved without entangling transformations [CKH98].

The QC presented in Tab. 5.1 was evolved on the GP system using the parameters shown in Tab. 5.2.

```
HAD     [1]
HAD     [0]
ORACLE
HAD     [1]
HAD     [0]
```

Table 5.1: QC found by the GP system in the 9th generation. See Fig. 5.1 for the graph representation of this circuit. The bracketed parameters denote the qubits the Hadamard gates are applied to. The oracle gate is applied to both qubits by definition.

Population size	500
No. of generations	200
Tournament size	8
Crossover probability	0.05
Creation probability	0.05
Mutation probability	0.90
Swap mutation probability	0.30×0.9
Grow mutation probability	0.30×0.9
Shrink mutation probability	0.20×0.9
Shrink2 mutation probability	0.20×0.9
No. of rotation angles	128
Max. no. of gates	200
Max. no. of oracle gates	5
Gate set	$\boldsymbol{H}$, $\boldsymbol{CNOT}$, $\boldsymbol{R}_x(2 \cdot \theta_l)$, $\boldsymbol{R}_y(2 \cdot \theta_l)$, $\boldsymbol{O}$

Table 5.2: Parameters of the GP system that evolved the QC depicted in Fig. 5.1. The rotation angle θ_l is specified by the integer $l \in \{0, 1, ..., 127\}$ via $\theta_l = -\pi + (l+1) \cdot \frac{2\pi}{128}$.

As indicated in Fig. 5.1 the circuit starts with the initial state $|\psi_{init}\rangle = |\underline{0}\rangle$. A superposition over all possible input states is created by applying Hadamard gates to each qubit. After a single oracle call the Hadamard gates are applied once more to each qubit:

$$|\psi_{fin}\rangle = \boldsymbol{H}^{\otimes 2}\, \boldsymbol{O}\, \boldsymbol{H}^{\otimes 2} |\underline{0}\rangle = \frac{1}{4} \sum_{i=0}^{3} \sum_{j=0}^{3} (\imath)^{x_i} (-1)^{\underline{i} \cdot \underline{j}} |\underline{j}\rangle.$$

The last term follows due to Eq. (3.14). The probability $prob\,(|\underline{0}\rangle\langle\underline{0}|)$ to measure the state $|\underline{0}\rangle$ is:

$$prob\,(|\underline{0}\rangle\langle\underline{0}|) = \left| \frac{1}{4} \sum_{i} (\imath)^{x_i} \right|^2 = \begin{cases} 1 & \text{if } X \in A \\ 0 & \text{if } X \in B \end{cases} .$$

Therefore, the problem can be solved by a single oracle call and a measurement of the two query qubits.

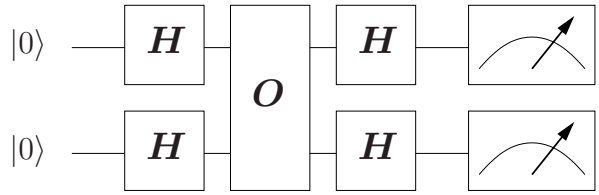

Figure 5.1: This exact QC distinguishes between blackboxes $X \in B$ and $X \in A$ by mapping the initial state $|\underline{0}\rangle$ to $|\underline{0}\rangle$ when $X \in A$. If $X \in B$ then the initial state is mapped to any state different from $|\underline{0}\rangle$.

This circuit can be generalized to an arbitrary number of qubits:

Proof:

The problem is to distinguish between blackboxes $X \in A$ whose entries are all equal and blackboxes $X \in B$ whose entries are all different.

To solve this problem one starts with the initial n-qubit state $|\psi_{init}\rangle = |\underline{0}\rangle$ by applying Hadamard gates to each qubit. After a single oracle call, Hadamard operations are applied once more to each qubit:

$$|\psi_{fin}\rangle = \boldsymbol{H}^{\otimes n}\,\boldsymbol{O}\,\boldsymbol{H}^{\otimes n}|\underline{0}\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}\left(e^{\imath\frac{2\pi}{2^n}}\right)^{x_i}(-1)^{\underline{i}\cdot\underline{j}}|\underline{j}\rangle.$$

Here we used the definition of the complex phase ξ in Eq. (4.5) and that a blackbox queried by n qubits contains 2^n elements. The probability to measure the state $|\underline{0}\rangle$ is:

$$prob\left(|\underline{0}\rangle\langle\underline{0}|\right) \quad = \quad \left|\frac{1}{2^n}\sum_{i=0}^{2^n-1}\left(e^{\imath\frac{2\pi}{2^n}}\right)^{x_i}\right|^2 = \begin{cases} 1 & \text{if } X \in A \\ 0 & \text{if } X \in B \end{cases}.$$

It follows that the problem can be solved by a single oracle call. □

The problem can easily be solved on a classical computer:

Proof:

To distinguish between blackboxes $X \in A$ and $X \in B$ one has to query the blackbox only twice. If both queries return the same answer then $X \in A$, otherwise $X \in B$. □

Therefore, this problem is not as interesting. Also, it can be solved by DJ's QA:

Proof:

Consider the rightmost bit in the binary decomposition of the blackbox entries: For a blackbox $X \in A$ this bit will have the same value for all entries. For a blackbox $X \in B$ one half of these entries will have the value 0, the remaining half the value 1. The problem can be solved by the DJ algorithm whose oracle gate encodes only this rightmost bit of a blackbox element. □

The above problem was already investigated and solved by Chi *et al.* [CKL01]. We modified it to see if the GP system is still able to find QCs that solve this new one.

5.2 A Special Case of the Hidden Subgroup Problem

The first instance of the problem we consider in this section is to distinguish between blackboxes $X \in A$ and $X \in B$ with

$$A = \{(a,a,a,a)\} \quad \text{and} \quad B = \{(a,a,b,b),(a,b,a,b),(a,b,b,a)\}, \tag{5.2}$$

here $a \neq b$ and $a, b \in \{0,1,2,3\}$.

5.2.1 A Probabilistic Quantum Circuit

We began our investigation using the oracle gates defined in Eq. (5.1). The QC of lowest error probability, returned by the GP system, was only slightly better than any classical probabilistic circuit. This QC was found with an old version of our GP system: Apart from the Hadamard gate $\boldsymbol{H}$, only rotation gates $\boldsymbol{R}(\theta_l)$ and $\boldsymbol{CNOT}$-gates were implemented:

$$\boldsymbol{R}(\theta_l) = \begin{pmatrix} \cos(\theta_l) & \sin(\theta_l) \\ -\sin(\theta_l) & \cos(\theta_l) \end{pmatrix}.$$

A hand-improved version of the best circuit found is shown in Fig. 5.2:

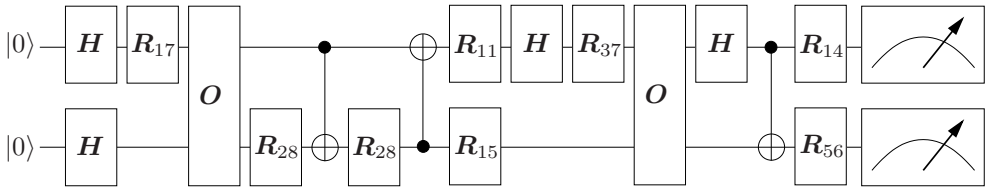

Figure 5.2: Probabilistic QC that is able to distinguish blackboxes $X \in A$ from blackboxes $X \in B$ with two oracle calls. $\boldsymbol{R}_l$ denotes a rotation gate whose rotation angle θ_l is specified by the integer $l \in \{0,1,...,63\}$ via $\theta_l = -\pi + (l+1) \cdot \frac{2\pi}{64}$.

In order to calculate the error probability of the QC one has to calculate the maximal misclassification probability for all blackboxes $X \in A \cup B$:

i) $X \in A \Rightarrow$ the probability that the QC answers $X \in A$ is 0.908.

ii) $X \in B \Rightarrow$ the probability that the QC answers $X \in B$ is 0.892.

Now we consider a classical circuit that has to solve this problem with only two oracle calls: If two distinct blackbox elements have the same value the circuit infers that the blackbox X belongs to set A. Therefore, a blackbox $X \in A$ will always be correctly classified. Nevertheless, also some blackboxes $X \in B$ will be classified to belong to set A. The probability that this happens is $1/3$. For a classical circuit one gets:

i) $X \in A \Rightarrow$ the probability that a classical circuit answers $X \in A$ is 1.0.

ii) $X \in B \Rightarrow$ the probability that a classical circuit answers $X \in B$ is $2/3$.

If the blackboxes X are taken equiprobable from the sets A and B then the misclassification rate of a QC is 0.1; the misclassification rate of a classical circuit is $0.1\overline{6}$. Hence, the QC is slightly better than any classical circuit.

Unfortunately, the scaling properties of this QC were not identifiable. In the next section we demonstrate that another set of oracle gates led to QCs whose scaling properties were apparent.

5.2.2 An Exact Quantum Circuit

A blackbox $X \in A \cup B$ with A and B defined in Eq. (5.2) returns the values $x_i \in \{0, 1, 2, 3\}$ if queried. The next oracle definition provides the only way to encode the blackbox elements into phase shifts and an output register according to Eq. (4.6):

$$\boldsymbol{O}|i\rangle|b\rangle = (-1)^{x_i^{(phase)}}|i\rangle|b \oplus x_i^{(xor)}\rangle. \tag{5.3}$$

Here the blackbox entries x_i are decomposed into the two binary values $x_i^{(phase)}$ and $x_i^{(xor)}$. Consider, for example, the blackbox $X = (0, 3, 0, 3)$: Decomposing the blackbox entries into two binary values returns the alternative representation $\tilde{X} = (00, 11, 00, 11)$:

$$\begin{aligned} x_0^{(phase)}x_0^{(xor)} &= 00, \quad x_1^{(phase)}x_1^{(xor)} = 11, \\ x_2^{(phase)}x_2^{(xor)} &= 00, \quad x_3^{(phase)}x_3^{(xor)} = 11. \end{aligned}$$

The oracle gate $\boldsymbol{O}_{(0,3,0,3)}$ has the following form:

$$\boldsymbol{O}_{(0,3,0,3)} = \begin{pmatrix} 1 & 0 & & & & & & \\ 0 & 1 & & & & & & \\ & & 0 & -1 & & & & \\ & & -1 & 0 & & & & \\ & & & & 1 & 0 & & \\ & & & & 0 & 1 & & \\ & & & & & & 0 & -1 \\ & & & & & & -1 & 0 \end{pmatrix}.$$

This approach requires an additional output qubit. Hence, the circuit needs three instead of two qubits. Nevertheless, this still is an advantage compared with the four qubits necessary for the standard encoding defined in Eq. (4.3).

It is possible to reduce the number of blackboxes that are to be tested. With the oracle defined in Eq. (5.3) the blackbox $X = (0, 0, 1, 1)$ and the blackbox $Y = (2, 2, 3, 3)$ are indistinguishable up to a global phase shift: It is sufficient to only test

X. Therefore, one has to test 2 instead of 4 blackboxes of set A and 18 instead of 36 blackboxes of set B. In comparison to the standard encoding the evaluation of a QC is reduced by the factor 2×2.

The shortest QC found by the GP system during 60 runs is the one shown in Tab. 5.3. Its graph representation can be found in Fig. 5.3. The parameters of the GP system are shown in Tab. 5.4.

In contrast to the probabilistic QC presented above, this circuit is an exact one whose structure shows much more regularity.

```
H        [1]
H        [2]
ORACLE
CNOT     [2 1]
Ry       [0 95]
CNOT     [1 2]
ORACLE
H        [2]
H        [1]
```

Table 5.3: QC found by the GP system in a total of 60 runs. It was found in the 166th generation of the first run. `Ry [0 95]` denotes that the one-qubit gate $\boldsymbol{R}_y(2 \cdot \theta_l)$ is applied to qubit 0. The angle θ_l is defined by: $\theta_l = -\pi + (l+1) \cdot \frac{2\pi}{128}$. One gets: $\theta_{95} = \pi/2$.

Population size	500
Max. no. of generations	5000
Tournament size	8
Crossover probability	0.05
Creation probability	0.05
Mutation probability	0.90
Swap mutation probability	0.30×0.9
Grow mutation probability	0.30×0.9
Shrink mutation probability	0.20×0.9
Shrink2 mutation probability	0.20×0.9
No. of rotation angles	128
Max. no. of gates	100
Max. no. of oracle gates	4
Gate set	$\boldsymbol{CNOT}$, $\boldsymbol{R}_x(2 \cdot \theta_l)$, $\boldsymbol{R}_y(2 \cdot \theta_l)$, $\boldsymbol{H}$, $\boldsymbol{O}$

Table 5.4: Parameters of the GP system that evolved the linear genome in Tab. 5.3. The angle θ_l is defined by: $\theta_l = -\pi + (l+1) \cdot \frac{2\pi}{128}$.

5.2.3 An Exact Quantum Algorithm

We are interested in developing QAs and not only QCs. Therefore, we tried to find the next problem instance for blackboxes with 8 elements. If the GP system returns

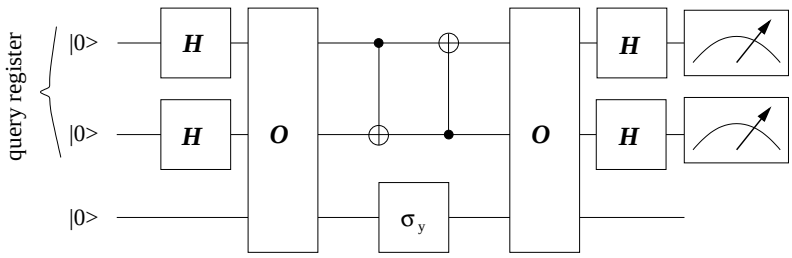

Figure 5.3: This circuit distinguishes between blackboxes $X \in A$ and $X \in B$ by mapping the initial state $|\psi_{init}\rangle = |00\rangle$ of the query register to the final state $|\psi_{fin}\rangle = |00\rangle$ if $X \in A$, otherwise the state $|00\rangle$ of the query register has a probability amplitude of zero. The gate $\boldsymbol{\sigma}_y$ is realized by $\boldsymbol{R}_y(\pi)$: $\boldsymbol{\sigma}_y = \imath \boldsymbol{R}_y(\pi)$.

a QC that solves this instance then it might be possible that an investigation of both circuits returns a scaling scheme.

Hence, the task is two-fold: On one hand we have to find the general structure of the problem so that the GP system returns QCs for its smallest instances. On the other hand we have to derive the general scaling mechanism by means of these circuits. This scaling mechanism provides a procedure to construct a uniform circuit family and thus the QA.

The elements x_i of blackboxes $X \in B$ defined in Eq. (5.2) meet the condition:

$$\underline{i} = \underline{j} \oplus \underline{s} \quad \Leftrightarrow \quad x_i = x_j, \quad i, j, s \in \{0, 1, 2, 3\}. \tag{5.4}$$

For the next problem instance we decided to modify this to:

$$\underline{i} = \underline{j} \oplus \underline{s} \quad \Rightarrow \quad x_i = x_j, \quad i, j, s \in \{0, 1, \ldots, 6, 7\}.$$

This modification implies that $x_i = x_j$ even if $\underline{i} \neq \underline{j} \oplus \underline{s}$. Nevertheless, for $i, j, s \in \{0, 1, 2, 3\}$ this modification is equal to the condition stated by Eq. (5.4). We used the values of blackboxes $X \in B$ with 4 elements to assign values to blackboxes $X' \in B'$ with 8 elements as indicated in Fig. 5.4. Therefore, the next instance of the problem

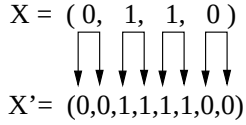

Figure 5.4: The assignment of the blackbox $X \in B$ is used to determine the elements of $X' \in B'$. In this example the elements of a blackbox with 8 elements are defined by: $\underline{i} = \underline{j} \oplus 001 \Rightarrow x_i = x_j$. For $X = (0, 1, 0, 1)$ one gets $X' = (0, 0, 1, 1, 0, 0, 1, 1)$.

is to distinguish between blackboxes $X' \in A'$ and $X' \in B'$:

$$A' = \{(a,a,a,a,a,a,a,a)\} \text{ and } B' = \left\{ \begin{array}{ll} (a,a,a,a,b,b,b,b), & (a,b,a,b,a,b,a,b), \\ (a,a,b,b,a,a,b,b), & (a,a,b,b,b,b,a,a), \\ (a,b,a,b,b,a,b,a), & (a,b,b,a,a,b,b,a), \\ (a,b,b,a,b,a,a,b) & \end{array} \right\}, \tag{5.5}$$

here $a \neq b$ and $a, b \in \{0,1,2,3\}$.

On a total of 60 runs the shortest exact QC was found in the 423rd generation of the third run. This QC is shown in Fig. 5.5. The settings of the GP system were identical to those shown in Tab. 5.4.

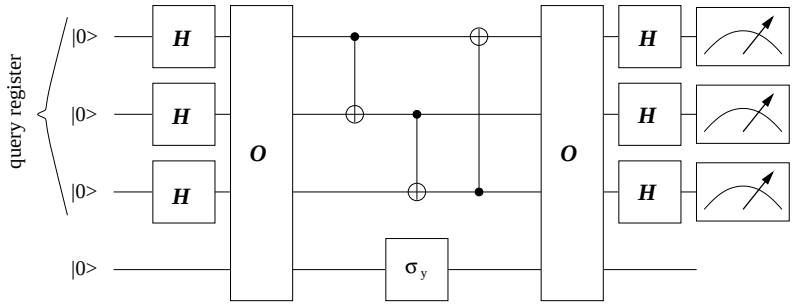

Figure 5.5: This QC distinguishes between blackboxes $X' \in A'$ and $X' \in B'$ by mapping the initial state $|\psi_{init}\rangle = |000\rangle$ of the query register to the final state $|\psi_{fin}\rangle = |000\rangle$ if $X' \in A'$, otherwise the state $|000\rangle$ of the query register has a probability amplitude of zero.

The construction scheme used above results in the following condition for blackbox elements x_i to be equal:

$$\left[(\underline{j} = \underline{i} \oplus \underline{s}) \text{ or } (\underline{j} = \underline{i} \oplus \underline{s'}) \text{ or } (\underline{j} = \underline{i} \oplus \underline{s} \oplus \underline{s'})\right] \Leftrightarrow x_i = x_j,$$

with $\underline{i}, \underline{j}, \underline{s}, \underline{s'} \in \mathcal{I}$ where $\mathcal{I} = \{\underline{0}, \underline{1}, \underline{2}, \underline{3}, \underline{4}, \underline{5}, \underline{6}, \underline{7}\}$ is the set of indices and $\underline{s} \neq \underline{s'}$. Together with the zero element $\underline{0}$ the elements $\underline{s}$, $\underline{s'}$ and $\underline{s} \oplus \underline{s'}$ form the group $K = \{\underline{0}, \underline{s}, \underline{s'}, \underline{s} \oplus \underline{s'}\}$ with $\oplus$ as group multiplication operation. Thus, the condition for elements x_i and x_j to be equal reads:

$$\underline{j} \in \underline{i} \oplus K \Leftrightarrow x_i = x_j,$$

where $\underline{i} \oplus K := \{\underline{i} \oplus \underline{k} : \underline{k} \in K\}$ denotes the coset of K. Hence, $x_i = x_j$ if and only if $\underline{i}$ and $\underline{j}$ are elements of the same coset. Using this concept, elements x_i of blackboxes $X' \in A'$ can be defined by:

$$x_i = x_j \Leftrightarrow \underline{j} \in \underline{i} \oplus K \text{ with } K = \mathcal{I}.$$

With this notation the two problems investigated are instances of a special case of the hidden subgroup problem defined in Tab. 5.5.

We demonstrate in Tab. 5.6 how this definition covers the problem instance defined in Eq. (5.5).

A special case of the hidden subgroup problem
One is given the finite Abelian group $\mathcal{I} = \{0,1\}^n$ of indices that has $\|\mathcal{I}\| = N = 2^n$ elements. Also one is given the subgroup $K \subset \mathcal{I}$. The group multiplication operation is the bitwise XOR operation ($\oplus$). Additionally one is given a blackbox $X = (x_0, x_1, \ldots, x_{N-1})$ with $x_i \in H$, $H = \{0,1,2,3\}$ and $\underline{i} \in \mathcal{I}$. One is promised that either $X \in A$ or $X \in B$. The two sets A and B are defined by:
$X \in A \quad \Leftrightarrow$ there exists only the subgroup $K = \mathcal{I}$ so that: $\forall i \neq j$ with $\underline{i}, \underline{j} \in \mathcal{I}: \; x_i = x_j \Leftrightarrow \underline{j} \in \underline{i} \oplus K$.
$X \in B \quad \Leftrightarrow$ there exists a subgroup $K \subset \mathcal{I}$ with $\|K\| = \|\mathcal{I}\|/2$ so that: $\forall i \neq j$ with $\underline{i}, \underline{j} \in \mathcal{I}: \; x_i = x_j \Leftrightarrow \underline{j} \in \underline{i} \oplus K$.
Decide which set X belongs to.

Table 5.5: Definition of our special case of the hidden subgroup problem.

Inspection of the QCs returned by the GP system on these two problem instances reveals a common functionality that is captured by the QA shown in Fig. 5.6. In what follows we will prove that this QA indeed solves this special case of the hidden subgroup problem:

Proof:

In this proof the state $|\underline{0}\rangle|0\rangle$ describes the initial $(n+1)$-qubit state: $|\underline{0}\rangle$ denotes the initial state of the n-qubit query register and $|0\rangle$ denotes the initial state of the single ancillary qubit (0th qubit).

The quantum gate $\boldsymbol{CNOT}_{l,m}$ denotes a $\boldsymbol{CNOT}$ gate as defined in Eq. (3.12). The lth qubit is the control and the mth qubit is the target qubit. The sequence of $\boldsymbol{CNOT}$ gates is abbreviated by $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \boldsymbol{CNOT}_{1,n}\boldsymbol{CNOT}_{2,1}\cdots\boldsymbol{CNOT}_{n-1,n-2}\boldsymbol{CNOT}_{n,n-1}. \tag{5.6}$$

We use the notation:

$$\boldsymbol{\Sigma}|\underline{i}\rangle = |\underline{i} \oplus \underline{\xi}(\underline{i})\rangle =: |\underline{\sigma}(\underline{i})\rangle,$$

where $\underline{\xi}(\underline{i})$ is uniquely determined by $\underline{i}$. Hence, $\underline{\sigma}(\underline{i})$ permutes the indices $\underline{i} \in \{0,1\}^N$.

Up to a global phase shift the effect of the one-qubit gate $\boldsymbol{\sigma}_y$ is described by:

$$\boldsymbol{\sigma}_y|b\rangle = (-1)^b|\bar{b}\rangle, \quad \text{with } b \in \{0,1\}.$$

The oracle gate is defined in Eq. (5.3), the n-qubit Hadamard gate is defined in Eq. (3.14).

Provided with these definitions we calculate the final quantum state obtained after applying the QA of Fig. 5.6 to the initial state $|\underline{0}\rangle|0\rangle$: After the first n Hadamard gates the initial state is mapped to:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_{i=0}^{2^n-1} |\underline{i}\rangle \right) |0\rangle.$$

Applying the first oracle to this state returns:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_{i=0}^{2^n-1} (-1)^{x_i^{(phase)}} |\underline{i}\rangle |x_i^{(xor)}\rangle \right).$$

The sequence of $\boldsymbol{CNOT}$ gates leaves us with:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_{i=0}^{2^n-1} (-1)^{x_i^{(phase)}} |\underline{\sigma}(\underline{i})\rangle |x_i^{(xor)}\rangle \right),$$

which, up to a global phase, is mapped by the $\boldsymbol{\sigma}_y$ gate to:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_{i=0}^{2^n-1} (-1)^{x_i^{(phase)}} |\underline{\sigma}(\underline{i})\rangle (-1)^{x_i^{(xor)}} |\bar{x}_i^{(xor)}\rangle \right) .$$

Once again the oracle gate is applied:

$$|\psi_e\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_{i=0}^{2^n-1} (-1)^{x_i^{(phase)} \oplus x_{\sigma(i)}^{(phase)}} |\underline{\sigma}(\underline{i})\rangle (-1)^{x_i^{(xor)}} |\bar{x}_i^{(xor)} \oplus x_{\sigma(i)}^{(xor)}\rangle \right) . \tag{5.7}$$

For blackboxes $X \in A$ one gets, up to a global phase shift:

$$|\psi_e\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\underline{\sigma}(\underline{i})\rangle \right) |1\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\underline{i}\rangle \right) |1\rangle.$$

The last equation follows because $\sigma(i)$ simply permutes the inputs i. After the final Hadamard gates one gets, up to a global phase shift, the final state $|\psi_f\rangle$:

$$|\psi_f\rangle = |\underline{0}\rangle|1\rangle.$$

Thus, if $X \in A$ then the query register is in the state $|\underline{0}\rangle$.

If $X \in B$ then one half of the blackbox's elements have a different value than the other half. Therefore, one has to distinguish two possibilities for the permutation $\sigma(i)$: Either $x_i = x_{\sigma(i)}$ or $x_i \neq x_{\sigma(i)}$. In the latter case one gets the following three possibilities:

a)

$$x_i \neq x_{\sigma(i)} \quad \Leftrightarrow \quad \left(x_i^{(phase)} \neq x_{\sigma(i)}^{(phase)} \text{ and } x_i^{(xor)} \neq x_{\sigma(i)}^{(xor)} \right)$$

b)

$$x_i \neq x_{\sigma(i)} \quad \Leftrightarrow \quad \left(x_i^{(phase)} = x_{\sigma(i)}^{(phase)} \text{ and } x_i^{(xor)} \neq x_{\sigma(i)}^{(xor)}\right)$$

c)

$$x_i \neq x_{\sigma(i)} \quad \Leftrightarrow \quad \left(x_i^{(phase)} \neq x_{\sigma(i)}^{(phase)} \text{ and } x_i^{(xor)} = x_{\sigma(i)}^{(xor)}\right)$$

Therefore, the sum over indices i in Eq. (5.7) is split into a sum over indices i with $x_i = x_{\sigma(i)}$ and indices i with $x_i \neq x_{\sigma(i)}$. One gets:

a)

$$|\psi_e\rangle = \frac{1}{\sqrt{2^n}} \left[\left(\sum_{i:x_i = x_{\sigma(i)}} (-1)^{x_i^{(xor)}} |\underline{\sigma(i)}\rangle \right) |1\rangle - \left(\sum_{i:x_i \neq x_{\sigma(i)}} (-1)^{x_i^{(xor)}} |\underline{\sigma(i)}\rangle \right) |0\rangle \right]$$

b)

$$|\psi_e\rangle = \frac{1}{\sqrt{2^n}} \left[\left(\sum_{i:x_i = x_{\sigma(i)}} (-1)^{x_i^{(xor)}} |\underline{\sigma(i)}\rangle \right) |1\rangle + \left(\sum_{i:x_i \neq x_{\sigma(i)}} (-1)^{x_i^{(xor)}} |\underline{\sigma(i)}\rangle \right) |0\rangle \right]$$

c)

$$|\psi_e\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_{i:x_i = x_{\sigma(i)}} (-1)^{x_i^{(xor)}} |\underline{\sigma(i)}\rangle - \sum_{i:x_i \neq x_{\sigma(i)}} (-1)^{x_i^{(xor)}} |\underline{\sigma(i)}\rangle \right) |1\rangle$$

In all three cases we have to show that in each sum there are as many states of phase $(-1)^{x_i^{(xor)}} = -1$ as states of phase $(-1)^{x_i^{(xor)}} = 1$. If this is the case then the final Hadamard operations map such a superposition to a new one where the zero-state $|\underline{0}\rangle$ of the query register vanishes.[1] Hence, blackboxes $X \in B$ become distinguishable from blackboxes $X \in A$.

Now we discuss the different cases separately:

a,b) In both cases the superposition $|\psi_e\rangle$ is split into states $|\underline{i}\rangle$ with $x_i = x_{\sigma(i)}$ and states $|\underline{j}\rangle$ with $x_j \neq x_{\sigma(j)}$. We will show that the part of the superposition that contains states $|\underline{j}\rangle$ with $x_j \neq x_{\sigma(j)}$ has as many states of phase $(-1)^{x_j^{(xor)}} = 1$ as states of phase $(-1)^{x_j^{(xor)}} = 0$. This is also true for the remaining part of the superposition over states $|\underline{i}\rangle$ with $x_i = x_{\sigma(i)}$ due to the definition of blackboxes $X \in B$.

Let us consider indices j with $x_j \neq x_{\sigma(j)}$: If one assumes that the number of indices j with $x_j^{(xor)} = 0$ is not equal to the number of indices j with $x_j^{(xor)} = 1$ then, after permuting the indices via σ, the total number of indices l with $x_l^{(xor)} = 0$ and indices k with $x_k^{(xor)} = 1$ changes. This is not possible as σ only permutes the indices. Hence, for $x_j^{(xor)} \neq x_{\sigma(j)}^{xor}$ one has as many states with $x_j^{(xor)} = 0$ as states $x_j^{(xor)} = 1$.

[1] The state $|\psi\rangle = |\underline{i}\rangle$ is mapped by any Hadamard gate to $|\psi\rangle = |\underline{0}\rangle \pm \ldots$; the state $|\psi\rangle = -|\underline{j}\rangle$ is mapped by any Hadamard gate to $|\psi\rangle = -|\underline{0}\rangle \pm \ldots$. Applying a Hadamard gate to $|\psi\rangle = |\underline{i}\rangle - |\underline{j}\rangle$ therefore returns a superposition where the zero-state $|\underline{0}\rangle$ vanishes. This holds true for any superposition over as many states of phase $+1$ as -1.

c) The phases of the states in the superposition $|\psi_e\rangle$ are determined by the values $x_i^{(xor)}$. By the definition of case c) we know that $x_i^{(xor)} = const.$ Therefore, one only has to show that the number of indices i with $x_i = x_{\sigma(i)}$ equals the number of indices $x_i \neq x_{\sigma(i)}$: The $\mathbf{\Sigma}$-gate performs XOR-operations $\sigma(\underline{i}) = \underline{i} \oplus \underline{\xi}(\underline{i})$. As $\xi(i)$ is uniquely determined by i, the $\mathbf{\Sigma}$-operation tests all $\underline{\xi}(\underline{i}) \in \{0,1\}^n$. Blackbox elements of blackboxes $X \in B$ are characterized by the condition $x_i = x_j$ with $\underline{j} = \underline{i} \oplus \underline{s}$ for one half of all $\underline{s} \in \{0,1\}^n$. Hence, the number of indices i with $x_i = x_{\sigma(i)}$ is equal to the number of indices j with $x_j \neq x_{\sigma(j)}$. □

Subgroup	Cosets	Conditions for the blackbox elements x_i
K_0	$\mathcal{I}$	$x_0 = x_1 = x_2 = x_3 = a$ $x_4 = x_5 = x_6 = x_7 = a;\quad a \in H$
K_1	$\{\underline{0},\underline{1},\underline{2},\underline{3}\}$, $\{\underline{4},\underline{5},\underline{6},\underline{7}\}$	$x_0 = x_1 = x_2 = x_3 = a$ $x_4 = x_5 = x_6 = x_7 = b;\quad a \neq b;\ a,b \in H$
K_2	$\{\underline{0},\underline{1},\underline{4},\underline{5}\}$, $\{\underline{2},\underline{3},\underline{6},\underline{7}\}$	$x_0 = x_1 = x_4 = x_5 = a$ $x_2 = x_3 = x_6 = x_7 = b;\quad a \neq b;\ a,b \in H$
K_3	$\{\underline{0},\underline{1},\underline{6},\underline{7}\}$, $\{\underline{2},\underline{3},\underline{4},\underline{5}\}$	$x_0 = x_1 = x_6 = x_7 = a$ $x_2 = x_3 = x_4 = x_5 = b;\quad a \neq b;\ a,b \in H$
K_4	$\{\underline{0},\underline{2},\underline{4},\underline{6}\}$, $\{\underline{1},\underline{3},\underline{5},\underline{7}\}$	$x_0 = x_2 = x_4 = x_6 = a$ $x_1 = x_3 = x_5 = x_7 = b;\quad a \neq b;\ a,b \in H$
K_5	$\{\underline{0},\underline{2},\underline{5},\underline{7}\}$, $\{\underline{1},\underline{3},\underline{4},\underline{6}\}$	$x_0 = x_2 = x_5 = x_7 = a$ $x_1 = x_3 = x_4 = x_6 = b;\quad a \neq b;\ a,b \in H$
K_6	$\{\underline{0},\underline{3},\underline{4},\underline{7}\}$, $\{\underline{1},\underline{2},\underline{5},\underline{6}\}$	$x_0 = x_3 = x_4 = x_7 = a$ $x_1 = x_2 = x_5 = x_6 = b;\quad a \neq b;\ a,b \in H$
K_7	$\{\underline{0},\underline{3},\underline{5},\underline{6}\}$, $\{\underline{1},\underline{2},\underline{4},\underline{7}\}$	$x_0 = x_3 = x_5 = x_6 = a$ $x_1 = x_2 = x_4 = x_7 = b;\quad a \neq b;\ a,b \in H$

Table 5.6: Second instance of our hidden subgroup problem: $\mathcal{I} = \{\underline{0},\underline{1},\underline{2},\underline{3},\underline{4},\underline{5},\underline{6},\underline{7}\}$, $H = \{0,1,2,3\}$, $K_0 = \mathcal{I}$, $K_1 = \{\underline{0},\underline{1},\underline{2},\underline{3}\}$, $K_2 = \{\underline{0},\underline{1},\underline{4},\underline{5}\}$, $K_3 = \{\underline{0},\underline{1},\underline{6},\underline{7}\}$, $K_4 = \{\underline{0},\underline{2},\underline{4},\underline{6}\}$, $K_5 = \{\underline{0},\underline{2},\underline{5},\underline{7}\}$, $K_6 = \{\underline{0},\underline{3},\underline{4},\underline{7}\}$ and $K_7 = \{\underline{0},\underline{3},\underline{5},\underline{6}\}$.

5.2.3.1 Comparison with other algorithms

Our QA solves this special case of the hidden subgroup problem with only two oracle calls. If $|\mathcal{I}| = 2^n$ then any exact classical algorithm needs at most $n+1$ blackbox calls.[2] A probabilistic classical algorithm can solve this problem using $k > 1$ calls

[2] The definition of the problem shows that a deterministic classical algorithm has to choose a subgroup $\mathrm{K} \subset \mathcal{I}$ with $|\mathrm{K}| = |\mathcal{I}|/2$. Then the algorithm has to calculate a coset $g \oplus \langle \mathrm{K} \rangle$ of the generators $\langle \mathrm{K} \rangle$ of K. The elements $k \in g \oplus \langle \mathrm{K} \rangle$ of this coset plus the neutral element 0 are used to query the blackbox for the elements x_k. If all elements are equal then the blackbox can still belong to set B, thus an additional element of a different coset has to be tested. If this query also returns the same answer then the blackbox belongs to set A, otherwise to B. With $|\mathcal{I}| = 2^n$ one has $k = \log_2(|\mathrm{K}|) = \log_2(2^{n-1}) = n-1$ generators, hence one has to call the blackbox $(n+1)$ times in the worst case.

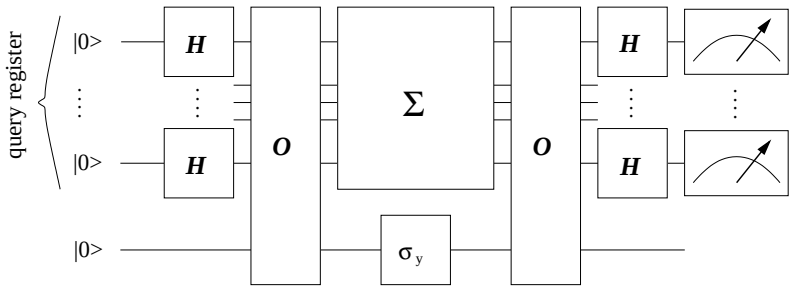

Figure 5.6: Generalized circuit for $(n+1)$ qubits. The upper n qubits encode $i \in \{0, 1, \ldots, 2^n - 1\}$. $\boldsymbol{\Sigma}$ abbreviates the gate sequence $\boldsymbol{CNOT}_{1,n}\boldsymbol{CNOT}_{2,1} \cdots \boldsymbol{CNOT}_{n-1,n-2}\boldsymbol{CNOT}_{n,n-1}$. $\boldsymbol{CNOT}_{l,m}$ denotes a $\boldsymbol{CNOT}$ gate with l the control and m the target qubit. This circuit distinguishes between blackboxes $X \in A$ and $X \in B$ by mapping the initial n-qubit state $|\psi_{init}\rangle = |0 \ldots 0\rangle$ of the query register to the final state $|\psi_{fin}\rangle = |0 \ldots 0\rangle$ if $X \in A$. Otherwise the state $|0 \ldots 0\rangle$ of the n-qubit query register has a probability amplitude of zero.

with an error probability of 2^{-k+1} if $|\mathcal{I}| \gg 1$.[3] Simon's QA can be generalized to a broader class of problems called Simon's hidden subgroup problems [BH97]. The corresponding generalized QA is a probabilistic one and can also be used to solve the problem presented here. Its error probability decreases with 2^{-k+1} in the number k of repetitions of this algorithm. Because this generalized algorithm calls the oracle only once, k repetitions of the algorithm correspond to k oracle calls. There exists an exact algorithm that solves Simon's hidden subgroup problem [BH97]. In our case it has to call the oracle three times.

It follows that the QA found with the help of the GP system is faster than any classical algorithm as well as any QA known to us.

5.2.4 Conclusion

The results stated above indicate that GP provides a useful tool to find new QAs. This conclusion is further supported by the results we are to discuss in the next section. There we will present formerly unknown QAs to solve the parity problem.

There are several possibilities to translate a problem to a blackbox problem. If we want to distinguish the set $A = \{2, 2, 2, 2\}$ from the set $B = \{1, 2, 2, 1\}$ then the blackboxes $X_1 = (2, 2, 2, 2)$ with $f(X_1) = 0$ and $X_2 = (1, 2, 2, 1)$ with $f(X_2) = 1$ provide a valid representation of this problem.

Alternatively, the set A can be represented by the blackboxes $X_{1,0} = (0, 0, 0, 0)$ and $X_{1,1} = (1, 1, 1, 1)$, the set B by the blackboxes $X_{2,0} = (1, 0, 0, 1)$ and $X_{2,1} = (0, 1, 1, 0)$. Here we used the binary decomposition of the elements to create the corresponding blackboxes, e.g., $X_{2,0}$ contains the rightmost digit of each element of B. Now the problem can be solved by two runs of the DJ algorithm (see Sec. 3.4.1). In the first run one queries the blackbox $X_{1,0}$ ($X_{2,0}$), in the second run one queries the blackboxes

[3] A blackbox that belongs to set B contains as many elements $x_i = a$ as elements $x_j = b$ where $a \neq b$ and $i \neq j$. Therefore, the probability that k different calls return the same answer decreases with 2^{k-1} for $|\mathcal{I}| >> 1$.

$X_{1,1}$ ($X_{2,1}$). If the DJ algorithm returns the answer "constant" in both cases then one knows that the set was A, otherwise B.

The latter approach requires some preprocessing to encode a single set by two blackboxes. These costs have to be taken into account to make a fair comparison between the two approaches presented here. Also, the latter blackbox problem does not fit in the blackbox model of computation introduced in Sec. 3.2.

5.3 The Parity Problem

The GP system was used to evolve QCs for two instances of the parity problem on single issue quantum computers as well as on ensemble quantum computers. In both cases we were able to derive formerly unknown QAs by means of theses QCs.

Sec. 5.3.1 formulates the parity problem. In Sec. 5.3.2 we will discuss the basic algorithm for single issue quantum computers. In Sec. 5.3.3 we will show that this algorithm can also be implemented on ensemble quantum computers. Sec. 5.3.4 presents the circuits returned by the GP system for ensemble quantum computers which allow for a further reduction in the number of oracle calls. Sec. 5.3.5 contains the experimental implementation on an NMR-QC and Sec. 5.3.6 draws conclusions.

5.3.1 Preliminaries

When the parity problem is formulated as a blackbox problem the desired property can be written as the Boolean function:

$$f(X) = x_0 \oplus x_1 \cdots \oplus x_{N-1}. \tag{5.8}$$

The first task of the GP system was to find a QC that is able to distinguish blackboxes $X \in A$ with $f(X) = 0$:

$$A = \{(a,a,a,a),(a,a,b,b),(a,b,a,b),(a,b,b,a)\},$$

from blackboxes $X \in B$ with $f(X) = 1$:

$$B = \{(a,a,a,b),(a,a,b,a),(a,b,a,a),(b,a,a,a)\},$$

for $a \neq b$ and $a, b \in \{0,1\}$.

The next problem was to find a QC that distinguishes blackboxes $X \in A$ with $f(X) = 0$:

$$A = \left\{ \begin{array}{llll} (a,a,a,a,a,a,a,a), & (a,a,a,a,a,a,b,b), & (a,a,a,a,a,b,b,a), & \dots \\ (a,a,a,a,b,b,b,b), & (a,a,a,b,b,b,b,a), & (a,a,a,b,b,b,a,b), & \dots \end{array} \right\},$$

from blackboxes $X \in B$ with $f(X) = 1$:

$$B = \left\{ \begin{array}{llll} (a,a,a,a,a,a,a,b), & (a,a,a,a,a,a,b,a), & (a,a,a,a,a,b,a,a), & \dots \\ (a,a,a,a,a,b,b,b), & (a,a,a,a,b,b,b,a), & (a,a,a,a,b,b,a,b), & \dots \end{array} \right\},$$

for $a \neq b$ and $a, b \in \{0,1\}$. Each of the sets A and B contains 64 elements.

The two QCs found by the GP system are shown in Fig. 5.7. They were sufficient to extract the scaling mechanism for an arbitrary number of qubits. The corresponding QA will be presented in the next section.

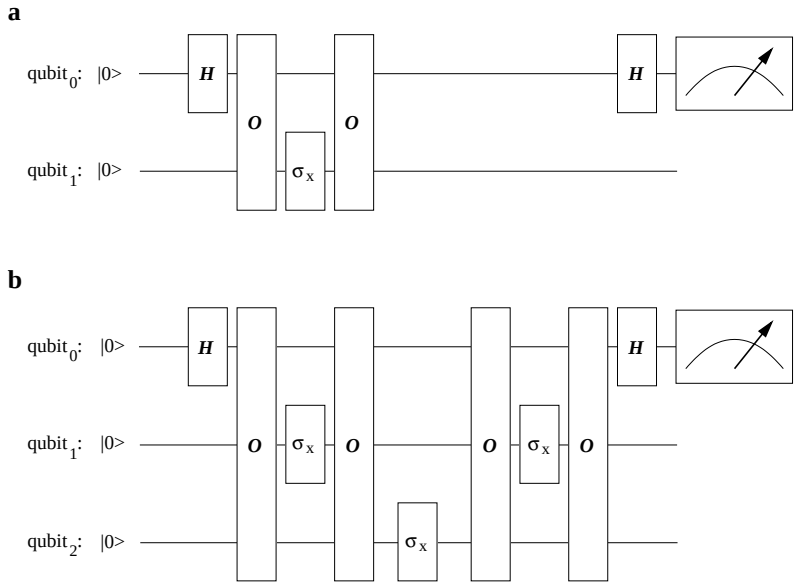

Figure 5.7: QCs found by the GP system for n=2 qubits (a) and n=3 qubits (b).

5.3.2 An Optimal Exact Quantum Algorithm

From now on we assume that the blackboxes, whose parity we want to calculate, contains N elements. Therefore, the corresponding oracle gate acts on N possible inputs i which are encoded into $n \geq \log_2 N$ qubits. If N is not a power of 2, the blackbox is extended with zeros, hence $N = 2^n$.

The gates used by the algorithm are the Hadamard operation $\boldsymbol{H}$, the NOT-operation $\boldsymbol{\sigma}_x$ and the n-qubit oracle gate $\boldsymbol{O}$. The formal definition of $\boldsymbol{O}$ can be found in Eq. (4.4).

In the simplest case of a one-qubit quantum register ($N = 2, n = 1$), which is equivalent to Deutsch's problem [Deu85], the parity of the string can be determined with a single oracle call: With the qubit initialized in the $|0\rangle$ state, we apply an oracle gate bracketed by two Hadamard gates. The resulting state of the quantum register is:

$$\begin{aligned} |\psi_{final}\rangle &= \boldsymbol{HOH}|0\rangle = \boldsymbol{HO}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \boldsymbol{H}\frac{1}{\sqrt{2}}((-1)^{x_0}|0\rangle + (-1)^{x_1}|1\rangle) \\ &= f(X)|1\rangle + (1 - f(X))|0\rangle, \end{aligned}$$

up to a global phase factor. Readout of the qubit shows the parity of X: even parity ($f(X) = 0$) results in a final state $|\psi_{final}\rangle = |0\rangle$ while odd parity results in $|\psi_{final}\rangle = |1\rangle$. The speedup by a factor of two, compared to the classical algorithm, results from the fact that the superposition determines whether the two bits are equal or opposite but does not differentiate between, e.g., the strings "00" and "11".

Now we prove that the QCs in Fig. 5.7 can be generalized to an arbitrary number of qubits.

Proof:

To generalize the QCs to strings of arbitrary length N we write the quantum register as:

$$|\psi\rangle = |\xi\rangle \otimes |\chi\rangle, \tag{5.9}$$

where $|\chi\rangle$ contains the single qubit that is used for readout, while $|\xi\rangle$ consists of the $n-1$ remaining qubits. All n qubits are first initialized into the $|0\rangle$ state; a Hadamard gate is then applied to the readout-qubit to create the superposition state:

$$|\psi_1\rangle = |0\cdots 0\rangle \otimes (\boldsymbol{H}|0\rangle) = \frac{1}{\sqrt{2}}(|00\cdots 0\rangle + |00\cdots 1\rangle).$$

If an oracle gate is applied to this state then it shifts the phase of each of the two components by π depending on the bit at position 0 or $N/2$ in X, respectively, being set. To take the other bits into account, we use repeated oracle calls with different inputs i. Since $\boldsymbol{O}$ does not modify the input vector $|\xi\rangle$, apart from the overall phase factor, we can generate the other inputs by subsequently flipping individual qubits. Fig. 5.7 summarizes the resulting algorithm for $n = 2$ and $n = 3$ qubits. In the $n = 2$ case the $|\xi\rangle$ component subsequently takes the values 0 and 1, in the $n = 3$ case it goes through $00 \to 10 \to 11 \to 01 \to 00$. The last step can be omitted but will be assumed here for the convenience of making the final state independent of the sequence of single qubit flips.

We summarize this sequence of N oracle calls alternating with σ_x equal to ***NOT*** operations with the unitary operator $\boldsymbol{U}_c$. Since its component operations $\boldsymbol{O}$ and σ_x are self-inverse and commute with each other one gets $\boldsymbol{U}_c = \boldsymbol{U}_c^{-1} = \boldsymbol{U}_c^{+}$. After this sequence of operations the state of the quantum register is:

$$\begin{aligned} |\psi_1\rangle &= \boldsymbol{U}_c \boldsymbol{H}^{(0)}|00...0\rangle = \frac{1}{\sqrt{2}}[(-1)^{x_0 \oplus x_2 \cdots \oplus x_{N-2}}|00...0\rangle \\ &+ (-1)^{x_1 \oplus x_3 \cdots \oplus x_{N-1}}|00...1\rangle]. \end{aligned}$$

The final Hadamard gate on the readout-qubit transforms this state into:

$$|\psi_{final}\rangle = \boldsymbol{H}^{(0)}|\psi_1\rangle = \begin{cases} |00\cdots 0\rangle \text{ if } f(X) = 0 \\ |00\cdots 1\rangle \text{ if } f(X) = 1 \end{cases}.$$

Hence, the state of the readout-qubit codes the parity $f(X)$ of the string X. □

The number of calls of the oracle gate ($N/2$) required by this algorithm coincides with the known lower bound [BBC+01, FGGS98]. Our algorithm is therefore optimal with respect to the number of oracle gates required, but also with respect to the number of additional gates, which are single qubit gates, independent of the size of the quantum register: If any of the ***NOT*** gates were omitted then two oracle gates would become adjacent to each other. According to Eq. (4.4) the oracle is its own inverse, so they could be eliminated from the algorithm thereby violating the lower bound.

Our algorithm requires the measurement of a single qubit, in contrast to the $N/2$ measurements used by the algorithm proposed by Beals *et al.* [BBC+01] and to the n measurements required by the algorithm of Farhi *et al.* [FGGS98].

In the next section we show that this parity algorithm can also be applied to ensemble quantum computers.

5.3.3 Application to an Ensemble Quantum Computer

To be able to discuss the operation of the algorithm on pure and mixed states within the same formal framework, we describe the state of the quantum register with a density operator. In most implementations, like in NMR-QCs, the initial state is the thermal state (see Sec 2.8.2):

$$\boldsymbol{\varrho}_{th} \approx \frac{1}{N}\left(\mathbf{1} - \boldsymbol{\mathcal{H}}\right) \approx \frac{1}{N}\left(\mathbf{1} - \sum_{i=0}^{n-1} \omega_i \boldsymbol{I}_z^{(i)}\right),$$

where we have set $\hbar/k_B T = 1$ and invoked the high-temperature approximation. Here $\boldsymbol{\mathcal{H}}$ denotes the Hamiltonian of the spin system, ω_i is the Larmor frequency of the ith spin (qubit) and $\boldsymbol{I}_z^{(i)}$ the corresponding spin operator.

Now we prove that the QA for single issue quantum computers of the last section also works for ensemble quantum computers:

Proof:

The initial Hadamard gate on the readout-qubit turns this state into:

$$\boldsymbol{\varrho} = \frac{1}{N}\left(\mathbf{1} - \omega_0 \boldsymbol{I}_x^{(0)} - \sum_{i=1}^{n-1} \omega_i \boldsymbol{I}_z^{(i)}\right). \tag{5.10}$$

The unity operator is time independent and does not contribute to any observable signal. The third term, which contains the thermal polarization of most of the spins, also does not contribute. We only need to consider the second term $\propto \boldsymbol{I}_x^{(0)}$. To compute the effect of the oracle gate on this term, we use the following decomposition:

$$\boldsymbol{I}_x^{(0)} = \frac{1}{2} \sum_{\xi=0}^{N/2-1} (|\underline{\xi}\rangle\langle\underline{\xi}| \otimes |0\rangle\langle 1| + H.c.),$$

where $\underline{\xi}$ stands for the binary representation of the integers ξ. The oracle gate turns this into:

$$\boldsymbol{O}\boldsymbol{I}_x^{(0)}\boldsymbol{O} = \frac{1}{2} \sum_{\xi=0}^{N/2-1} (-1)^{x_{2\xi} \oplus x_{2\xi+1}} (|\underline{\xi}\rangle\langle\underline{\xi}| \otimes |0\rangle\langle 1| + H.c.). \tag{5.11}$$

Like in the single-instance case, we cycle the system through all possible oracle inputs by applying the sequence $\boldsymbol{U}_c$ of oracle gates and bit-flip operations σ_x. Each term in the above sum then acquires the same phase factor:

$$\boldsymbol{U}_c \boldsymbol{I}_x^{(0)} \boldsymbol{U}_c = \frac{1}{2}(-1)^{f(X)} \sum_{\xi=0}^{N/2-1} \left(|\underline{\xi}\rangle\langle\underline{\xi}| \otimes |0\rangle\langle 1| + H.c.\right) = (-1)^{f(X)} \boldsymbol{I}_x^{(0)}.$$

By measuring the sign of the resulting spin-polarization of the readout-qubit we can directly determine the parity of the string in a single measurement. □

To our surprise the GP system returned QCs for ensemble quantum computers that need fewer oracle calls then the algorithm presented here and thus beat the lower bounds established by Beals *et al.* [BBC+01]. Investigation of the two QCs presented in Fig. 5.8 made it possible to derive the functionality and thus the scaling mechanism. The corresponding QA and its functionality is the topic of the next section.

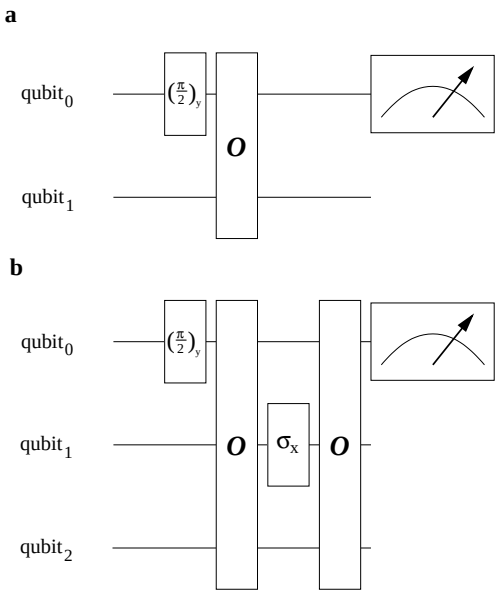

Figure 5.8: Mixed state parity circuits found by the GP system for $n = 2$ (a) and $n = 3$ (b) qubits. The $(\pi/2)_y$-gate corresponds to the rotation $\boldsymbol{R}_y(\frac{\pi}{2}) = e^{-i(\pi/4)\sigma_y} = 1/\sqrt{2}\,(\mathbf{1} - i\sigma_y)$. The measurement gate symbolizes a measurement of the magnetization along the x-axis.

5.3.4 Speed-up for Ensemble Quantum Computers

The determination of the parity requires at least $N/2$ oracle calls on a single issue quantum computer. Nevertheless, a modified algorithm to be run on an ideal and noiseless ensemble quantum computer can determine the parity by a single oracle call.

This can be seen by calculating the expectation value of the observable $\boldsymbol{I}_x^{(0)}$ for the state Eq. (5.11) of the quantum register after the first call of the oracle gate:

$$tr[\boldsymbol{I}_x^{(0)}\boldsymbol{O}\boldsymbol{I}_x^{(0)}\boldsymbol{O}] = \frac{1}{2}\sum_{\xi=0}^{N/2-1}(-1)^{x_{2\xi}\oplus x_{2\xi+1}}.$$

For even parity the sum can reach extremal values of $\pm N/2$, for odd parity they

are $\pm(N/2-2)$. The measured values are:

$$\langle \boldsymbol{I}_x^{(0)} \rangle_{f(X)=0} = \frac{r\omega_0}{2N} \quad \text{for } r \in \{-\frac{N}{2}, -\frac{N}{2}+4, \cdots \frac{N}{2}\} .$$
$$\langle \boldsymbol{I}_x^{(0)} \rangle_{f(X)=1} = \frac{s\omega_0}{2N} \quad \text{for } s \in \{-\frac{N}{2}+2, \cdots \frac{N}{2}-2\} . \tag{5.12}$$

A single call to the oracle gate thus allows one to determine the parity by measuring the expectation value of $\boldsymbol{I}_x^{(0)}$, provided the resolution of this measurement is high enough to distinguish between neighboring values.

This separation between neighboring values decreases with the length N of the string - i.e., exponentially with the number of qubits. The scheme is therefore not scalable for large systems. But even if the separation becomes too small to be resolved by the measurement, it remains possible to generate an exponential speedup over the single issue quantum computer at the cost of a correspondingly higher demand on the precision of the readout.

The two cases that we have considered so far, using $N/2$ and a single oracle call, respectively, can be considered extreme cases of a series of algorithms that require 2^{n-k-1} calls of the oracle gate. This corresponds to a speedup by 2^k compared to the single issue quantum computer as will be shown below.

Proof:

We subdivide the address register (5.9) into three parts:

$$|\psi\rangle = |\mu\rangle \otimes |\nu\rangle \otimes |\chi\rangle,$$

where $|\chi\rangle$ is again the single readout-qubit, while $|\xi\rangle = |\mu\rangle \otimes |\nu\rangle$ represents the remaining $n-1$ qubits. If the number of qubits in $|\nu\rangle$ is k, $|\mu\rangle$ contains only $n-k-1$ qubits.

We now restrict the number of oracle calls to all possible combinations of the qubits in $|\mu\rangle$ - i.e., 2^{n-k-1}. The relevant term:

$$\boldsymbol{I}_x^{(0)} = \frac{1}{2} \sum_{\mu=0}^{2^{n-k-1}-1} \sum_{\nu=0}^{2^k-1} \left(|\underline{\mu\nu}0\rangle\langle\underline{\mu\nu}1| + H.c. \right),$$

in the density operator (5.10) is then transformed into:

$$\boldsymbol{U}_c \boldsymbol{I}_x^{(0)} \boldsymbol{U}_c = \frac{1}{2} \sum_{\nu=0}^{2^k-1} \left[\left(\prod_{\mu=0}^{2^{n-k-1}-1} (-1)^{x_{\underline{\mu\nu}0} \oplus x_{\underline{\mu\nu}1}} \right) \times \sum_{\mu=0}^{2^{n-k-1}-1} \left(|\underline{\mu\nu}0\rangle\langle\underline{\mu\nu}1| + H.c. \right) \right],$$

where $\boldsymbol{U}_c = \boldsymbol{U}_c^{-1}$ represents the sequence of 2^{n-k-1} oracle and NOT gates. Calculating the expectation value for this state, in analogy to Eq. (5.12), we find:

$$tr[\boldsymbol{I}_x^{(0)} \boldsymbol{U}_c \boldsymbol{I}_x^{(0)} \boldsymbol{U}_c] = 2^{n-k-2} \sum_{\nu=0}^{2^k-1} \left(\prod_{\mu=0}^{2^{n-k-1}-1} (-1)^{x_{\underline{\mu\nu}0} \oplus x_{\underline{\mu\nu}1}} \right).$$

Similar to the results from the single oracle call, the expectation value for $\boldsymbol{I}_x^{(0)}$ depends on the parity $f(X)$:

$$\langle \boldsymbol{I}_x^{(0)} \rangle_{f(X)=0} = r\omega_0 2^{-k-2} \quad \text{for } r \in \{-2^k, -2^k+4, \cdots 2^k\}.$$
$$\langle \boldsymbol{I}_x^{(0)} \rangle_{f(X)=1} = s\omega_0 2^{-k-2} \quad \text{for } s \in \{-2^k+2, \cdots 2^k-2\}.$$

Expectation values indicating opposite parities are thus separated by:

$$|\langle \boldsymbol{I}_x^{(0)} \rangle_{f(X)=0} - \langle \boldsymbol{I}_x^{(0)} \rangle_{f(X)=1}| \geq \omega_0 2^{-k-1}.$$

The minimal separation therefore decreases exponentially with the number k of omitted address qubits, or linearly with the number of oracle calls saved. □

The algorithm proposed by X. Miao shows a similar exponential decrease in the difference of the signal strength necessary to decide the parity problem [Mia01]. In contrast to Miao's approach we do not require non-unitary quantum operations. Since our algorithm works directly with the thermal mixed state the signal strength suffers no exponential decrease if the number of qubits increases; this is similar to the modified DJ algorithm proposed by Myers *et al.* [MFGM01].

Recently, Arvind *et al.* criticized the ensemble version of the DJ algorithm proposed by Myers *et al.* to simply make use of the massive classical parallelism an ensemble computer provides [AC03]. They show that the ensemble:

$$\boldsymbol{\varrho}_{init} = \frac{1}{2^n} \sum_{i=0}^{2^n-1} |\underline{i}\rangle\langle\underline{i}| \otimes |0\rangle\langle 0|,$$

of an $n+1$ qubit system, with the first n query qubits in a fully mixed state and the readout-qubit in a pure state, can easily be used to explain the functionality of the algorithm proposed by Myers *et al.* [MFGM01].

Proof:

A standard oracle maps $\boldsymbol{\varrho}_{init}$ to:

$$\boldsymbol{\varrho}_{fin} = \frac{1}{2^n} \sum_{i=0}^{2^n-1} |\underline{i}\rangle\langle\underline{i}| \otimes |x_i\rangle\langle x_i|.$$

The information of the blackbox is contained in the readout-qubit whose reduced density matrix is:

$$\boldsymbol{\varrho}_{out} = \frac{1}{2^n} \sum_{i=0}^{2^n-1} |x_i\rangle\langle x_i|.$$

Measurement of $\boldsymbol{\sigma}_z$ returns:

$$\langle \boldsymbol{\sigma}_z \rangle = \begin{cases} \pm 1 & \text{if } X \in A \\ 0 & \text{if } X \in B \end{cases}.$$

The definition of the sets A and B for the DJ problem can be found in Sec. 3.4.1. Arvind *et al.* showed that this mechanism can be mimicked easily by a classical ensemble [AC03]. □

This kind of criticism does not hold for our ensemble QA because it also works for an ensemble of pure states $\boldsymbol{\varrho}$:

$$\boldsymbol{\varrho} = \boldsymbol{H}^{\otimes n}|0...0\rangle\langle 0...0|\boldsymbol{H}^{\otimes n} = \frac{1}{N}\sum_{i,j=0}^{N-1}|i\rangle\langle j|.$$

Our ensemble QA exploits the ability of ensemble quantum computers to distinguish non-orthogonal states. This will be discussed in more detail in Sec 5.3.6.

5.3.5 Experimental Implementation

We implemented the two-qubit version ($n = 2, N = 4$) of the exact parity algorithm as well as the reduced ensemble algorithm with $k = 1$ on an NMR-QC, using the spins of the ^{1}H and ^{13}C nuclei in a carbon-13 labeled chloroform molecule ($CHCl_3$) whose Hamiltonian is of the form (see Sec. 2.8.2):

$$\boldsymbol{\mathcal{H}} = -\omega_0^{(H)}\boldsymbol{I}_z^{(H)} - \omega_0^{(C)}\boldsymbol{I}_z^{(C)} + 2\pi J\,\boldsymbol{I}_z^{(H)}\boldsymbol{I}_z^{(C)}.$$

Here $\omega_0^{(H)}$ and $\omega_0^{(C)}$ denote the Larmor frequencies of the nuclear spins and J the strength of the scalar coupling between them.

We use a resonant rotating frame, where $\omega_0^{(H)} = \omega_0^{(C)} = 0$. All experiments were performed at room temperature on a home-built NMR spectrometer with a ^{1}H operating frequency of 360 MHz.

The exact version of the parity algorithm, which needs two oracle calls, was implemented as shown in Fig. 5.7a. The first Hadamard gate $\boldsymbol{H}$ was replaced by the pseudo-Hadamard operation $\mathbf{h}$ which corresponds to a $(\pi/2)_y$ rotation of the corresponding qubit around the $\boldsymbol{e}_y$-axis. The final Hadamard gate $\boldsymbol{h}^{-1}$ then cancels with the readout pulse that would otherwise be required to convert the final state into observable $\boldsymbol{I}_x^{(C)}$ magnetization. The readout (of transverse magnetization) therefore starts immediately after the last oracle gate. As an additional simplification we omitted the last bit reversal of the second qubit, which does not affect the readout-qubit.

The $\boldsymbol{\sigma}_x$-operation (NOT-gate) was realized by a $(\pi)_x$-pulse. The oracle gate $\boldsymbol{O}$ that represents the black-box $X = (x_0, x_1, x_2, x_3)$ has the matrix representation:

$$\boldsymbol{O} = \begin{pmatrix} (-1)^{x_0} & & & \\ & (-1)^{x_1} & & \\ & & (-1)^{x_2} & \\ & & & (-1)^{x_3} \end{pmatrix}.$$

The oracle gate for $X = (0, 0, 0, 1)$ can be realized by the pulse-sequence $\tau - (\pi/2)_{-z}^{C} - (\pi/2)_{-z}^{H}$. Here $\tau = 1/(2J)$ denotes the time of a free evolution period where the system evolves under the scalar spin-spin coupling. With $J = 215$ Hz one gets $\tau = 2.326$ ms.

The $(\theta)_{\pm z}$ rotations cannot be implemented directly by radio frequency pulses. They were realized by the composite pulse-sandwich $(\pi/2)_x - (\theta)_{\pm y} - (\pi/2)_{-x}$ [FFL81].

Similar sequences were determined for the other 15 oracle gates (see Appendix B). The resulting oracle gates are pairwise equivalent, modulo an overall phase factor, for strings with inverted bit values. As an example, compare the matrix representations

for $X = (0,0,0,0)$ and $X = (1,1,1,1)$, which correspond to $\pm\mathbf{1}$. Clearly, the overall phase factor does not affect the measured result. This ambiguity of the oracle gates is not critical for our application since the corresponding string pairs always have the same parity.

The pseudo-pure state necessary for the pure state algorithm was realized via temporal averaging [KCL98] - i.e., by adding up the spectra of three experiments in which the populations of the states $|01\rangle$, $|10\rangle$ and $|11\rangle$ were cyclically permuted (see Sec. 2.8.4).

The free induction signals of the carbon nuclei measured at the end of each parity algorithm was Fourier transformed and displayed in Fig. 5.10 for all possible strings with $N = 4$.

The uppermost trace shows, as a reference, the spectrum obtained by applying a readout pulse directly to the thermal equilibrium state. The two resonance lines correspond to the two spin orientations of the second (^{1}H) spin, which are almost equally populated in thermal equilibrium. The other traces represent the Fourier transformed free induction signals measured after applying the parity algorithm for the strings indicated to the pseudo-pure state $|00\rangle$.

According to the theoretical result we expect the sign of the ^{13}C signal to represent the parity of the string. This agrees with the experimental observation where the signal for the even parity strings is positive while the signal for the odd parity strings is negative.

In the pure state algorithm the second qubit is always in a definite state: $|0\rangle$ in the algorithm discussed in Sec. 5.3.2, $|1\rangle$ if the final ***NOT*** operation is omitted. Accordingly, only one of the two ^{13}C resonance lines has a non-vanishing amplitude.

As discussed in Sec. 5.3.3 the algorithm can also be applied to mixed states, thus eliminating the need to prepare a pseudo-pure state and avoiding the corresponding reduction of signal strength. We do not discuss the corresponding measurements here but proceed directly to the reduced version where the number of oracle calls is reduced to one (k=1). Fig. 5.9 shows the required sequence of gate operations.

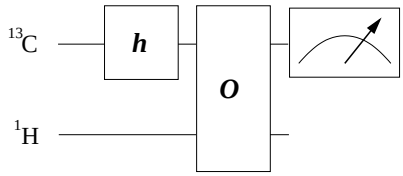

Figure 5.9: Ensemble QC for two qubits that uses a single call of the oracle gate.

Instead of the general Hadamard gate $\boldsymbol{H}$ we again use the pseudo-Hadamard gate $\boldsymbol{h}$ - i.e., a $(\pi/2)_y$ pulse. The oracle gate is the same as in the pure state case.

For the results of the reduced mixed state algorithm we only present the measurement results of $\langle \boldsymbol{I}_x^{(0)} \rangle$ at $t = 0$ - i.e., immediately after the end of the oracle gate. The results shown in Tab. 5.7 are $\pm(3.46 \pm 0.36)$ for the even parity strings and $\pm(0.24 \pm 0.07)$ for odd parity strings in good agreement with the theoretical predictions of Eq. (5.12).

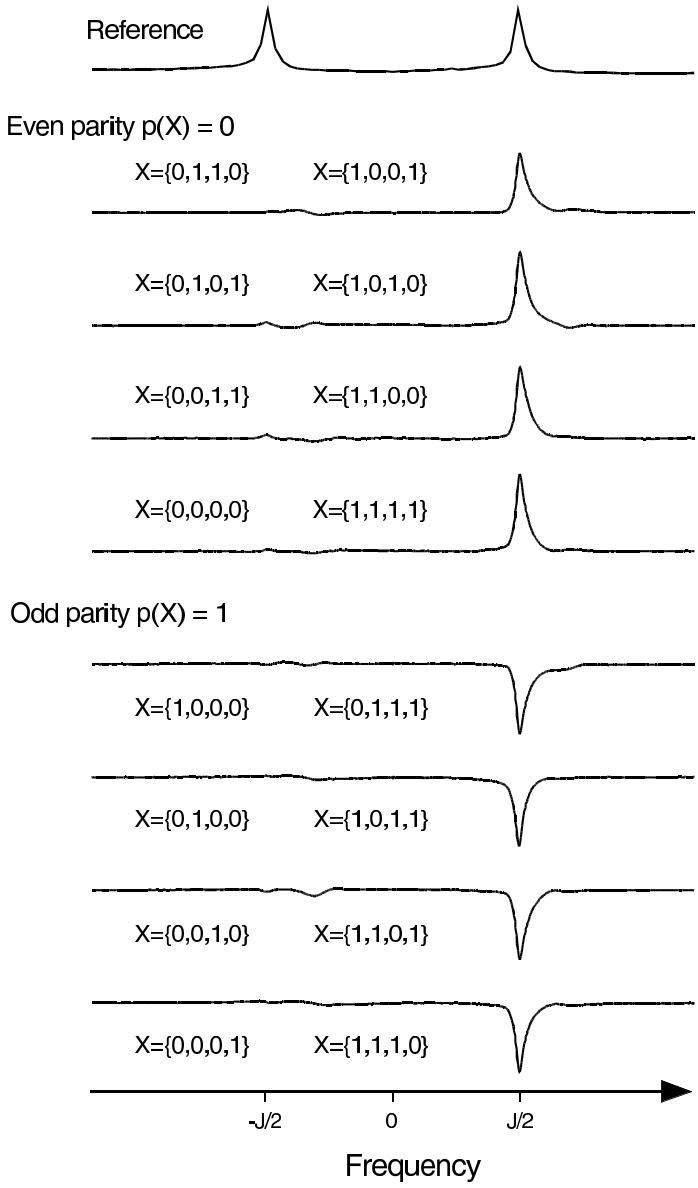

Figure 5.10: Experimental results for the pure state QC shown in Fig. 5.7. The uppermost trace shows the real part of the carbon spectrum after a readout pulse applied to the system in thermal equilibrium. The remaining spectra show the real part of the carbon spectrum after completion of the exact parity algorithm on the effectively pure initial state $|00\rangle$. The frequency is relative to 90.533504 MHz.

$f(X)=0$	$\langle I_x^{(0)}\rangle(t=0)$	$f(X)=1$	$\langle I_x^{(0)}\rangle(t=0)$
$X=(0,0,0,0)$	3.84	$X=(0,0,0,1)$	-0.29
$X=(0,0,1,1)$	-3.29	$X=(0,0,1,0)$	0.18
$X=(0,1,0,1)$	3.77	$X=(0,1,0,0)$	-0.32
$X=(0,1,1,0)$	-2.95	$X=(1,0,0,0)$	0.16

Table 5.7: Experimental results of $\langle \boldsymbol{I}_x^{(0)}\rangle$ for the ensemble QC of Figure 5.9. The numerical values are in arbitrary units.

5.3.6 Conclusion

We have introduced a family of QAs that solve the parity problem with an optimal number of quantum gate operations. It uses the blackbox scheme introduced by Beals *et al.* to represent the binary strings as oracle gates [BBC+01]. In agreement with the lower bound established by Beals *et al.* our algorithm uses $N/2$ calls of the oracle gate, a factor of two less than the best classical algorithms. This reduction, compared to the classical case, can be attributed to quantum parallelism since the input state to the oracle gate is a superposition of two basis states.

A further reduction of the number of oracle calls is possible when an ensemble quantum computer is used instead of a single issue quantum computer. In this case the number of calls can be reduced by a factor $2^k < N$ at the expense of a smaller separation between the measurement values that indicate even/odd parity. This additional speed-up requires parallel operation of many nominally identical quantum systems since a single system cannot provide the result in a single run.

The reduction of the number of oracle calls below the lower bound of $N/2$ is linked to the fact that ensemble quantum computers are able to distinguish non-orthogonal states as mentioned by Dorai *et al.* [DAK01]. Such non-orthogonal states are the result of probabilistic QAs.

To demonstrate the probabilistic nature of the ensemble algorithm we modify the ensemble QC in Fig. 5.8a. For convenience we add an additional qubit, a controlled-σ_x gate and two Hadamard gates. This makes it possible to replace the measurement of the magnetization by a projective measurement in the computational basis (see Fig. 5.11).

The ensemble QC in Fig. 5.8a returns $\langle \boldsymbol{I}_x^{(0)}\rangle = \pm max$ for blackboxes X with $f(X)=0$. Accordingly, on the single issue quantum computer in Fig. 5.11, the additional qubit either is in the state $|0\rangle$ or in the state $|1\rangle$. For blackboxes X with $f(X)=1$ the ensemble QC returns $\langle \boldsymbol{I}_x^{(0)}\rangle = 0$. In this case, on the single issue quantum computer, the additional qubit is in the state $1/\sqrt{2}\,(|0\rangle \pm |1\rangle)$.

Hence, a single run of the QC on a single issue quantum computer does not reveal any useful information. Nevertheless, if several runs return different measurement results then one knows that $f(X)=1$. If, on the other hand, $l \in \mathbb{N}$ runs return equal results then one has an error probability of 2^{-l+1} in claiming that $f(X)=0$.

The QC in Fig. 5.11 provides a useful probabilistic QC despite the fact that a single run does not better than guessing. This is in contrast to classical probabilistic algorithms. There one only considers those algorithms to be useful that have an error probability less then 1/2 (see Sec. 3.1.4). Our QC is not covered by this class of

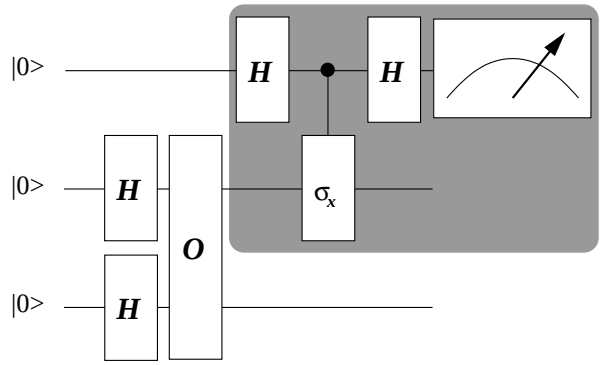

Figure 5.11: Modified instance of the ensemble QC in Fig. 5.8a.

probabilistic algorithms because its error probability is exactly 1/2.

6 Discussion and Outlook

The successful development of two formerly unknown QAs presented in Chap. 5 demonstrates the usefulness of evolutionary methods in generating new QAs.

One might be tempted to consider these results to be irrelevant for real-world problems as our GP system is only able to evolve blackbox quantum algorithms. Indeed, the blackbox model of computation is a much more restricted model than that introduced by Turing. Nevertheless, it has to be emphasized that QAs that solve real-world problems like integer factoring are based on the possibility to reduce the factoring problem to the problem of order-finding that is most naturally stated in the blackbox model of computation. Using the blackbox model it is also possible to show that there is an exponential gap in the query complexity of quantum as compared to classical algorithms. Thus, the somewhat artificial character of such problems does by no means indicate that they have no relevance. This substantiates our opinion that the investigation of blackbox algorithms is valuable, particularly with regard to the fact that within this model classical and QAs can be easily compared.

We restricted our investigations to the blackbox model of computation where the Boolean property $f(X)$ of a blackbox X is to be computed with a minimal number of blackbox (oracle) calls. Our GP system only allows to evolve QDTs that, after a sequence of oracle and quantum gates, reveal the property of the blackbox by a final measurement (see Sec. 4.2). This is similar to the approach used by Beals *et al.* in proving lower bounds for the parity problem [BBC+01]. Surprisingly, the GP system returned QCs that beat this lower bound on ensemble quantum computers. The error probability of these QCs on a single issue quantum computer is 1/2 (see Sec. 5.3.6). Nevertheless, running such a circuit several times on the same initial state makes it possible to solve the parity problem with an error probability that decreases exponentially in the number of repetitions.

This result indicates that the confinement to QDTs as defined in Eq. (3.6) that reveal the property of a blackbox by a final measurement does not fully explore the possibilities of quantum computation. Speaking about final measurements we do not mean intermediate measurements used to conditionally control subsequent quantum gates. Such measurements can always be replaced by unitary quantum operations. Therefore, they are covered by the definition of the QDT [NC00]. QAs like Simon's algorithm use several independent runs of a QA to collect measurement results for a further classical treatment. At the current stage our GP system wouldn't be able to evolve such hybrid algorithms. As far as we know such algorithms haven't been investigated by other authors [SBBS99, LB03b, Spe04] who use GP to design QAs, either. Hence, one has to elaborate in what respect the GP system has to be extended in order to evolve hybrid algorithms.

Also, it is worthwhile not only to consider worst-case query complexities but also average-case query complexities. This can be achieved by implementing controlled oracle gates that weren't used in the investigations presented here.

We emphasize that the most sensible part of setting up a GP system to aid the

development of QAs is designing the fitness function. The requirement we consider to be the most important is the continuity of the fitness function in the step-length of genetic operators as discussed in Sec. 4.4. The fitness function used by us only partly fulfills this condition, nevertheless, the results obtained are very promising. Therefore, designing a fitness function that hampers the search process as less as possible seems to be essential.

GP provides a valuable tool to develop new QAs when small problem instances are sufficient to derive the general scaling mechanism. This is a bold assumption as it is possible that some QAs become more efficient compared to classical ones only for large problem instances. According to the results presented in Sec. 4.3, GP will not be of any help in such cases. Nevertheless, it is quite probable that the set of algorithms is wide enough to provide plenty of algorithms that remain to be discovered using GP.

A Parity Algorithms

We present the two QAs suggested by Beals *et al.* [BBC+01] and Farhi *et al.* [FGGS98] to solve the parity problem. In Sec. 5.3.2 these two QAs were compared to the QA developed by us with the help of GP.

Applying the blackbox model of computation the parity problem reads like follows:

Parity Problem

One is given a blackbox $X = (x_0, x_1, \ldots, x_{N-1})$ of N Boolean variables x_i such that on input $i \in \mathcal{I}$ with $\mathcal{I} = \{0, 1 \ldots N-1\}$ the blackbox returns the corresponding value of x_i. One has to decide if $X \in A$ ($f(X) = 0$) or $X \in B$ ($f(X) = 1$):

i) $X \in A \quad \Leftrightarrow \quad$ The number of values $x_i = 1$ is even.

ii) $X \in B \quad \Leftrightarrow \quad$ The number of values $x_i = 1$ is odd.

The Boolean function $f(X)$ that represents the parity of a blackbox X can be calculated by $f(X) = x_{N-1} \oplus \cdots \oplus x_1 \oplus x_0$. Note that the function $f(X)$ is total, which means that this property is defined for all blackboxes X. Thus, as mentioned in Sec. 3.2.2, the quantum speed-up is at most polynomial. For the parity problem it is even known that QAs, exact as well as probabilistic, cannot achieve a speed-up of more than a factor of 2. Now we present two QAs that reach this lower bound.

A.1 Parity Algorithm proposed by Beals

The QA proposed by Beals *et al.* [BBC+01] makes use of the fact that the XOR operation (denoted by $\oplus$) on two Boolean variables x_l and x_m of X can be calculated by a QA with only one oracle call [CEMM97].[1] To see how this speed-up comes about one starts in the superposition $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|l\rangle + |m\rangle)$. Using the definition of the oracle gate in Eq. (3.4.1) one gets:

$$\boldsymbol{O}|\psi_0\rangle = \frac{1}{\sqrt{2}}\left((-1)^{x_l}|l\rangle + (-1)^{x_m}|m\rangle\right) = \frac{(-1)^{x_l}}{\sqrt{2}}\left[|l\rangle + (-1)^{x_l \oplus x_m}|m\rangle\right].$$

As global phases cannot be measured one can ignore the factor $(-1)^{x_l}$ on the right side of the last equation. Nevertheless, local phases can be detected and therefore a measurement would distinguish $x_l \oplus x_m = 0$ from $x_l \oplus x_m = 1$.

[1] It is to be emphasized that, as mentioned in [BBC+01], the XOR-operation and its negation are the only two of all 16 connectives on 2 variables where quantum computers provide a speed-up over classical computations.

Knowing that on a quantum computer it is possible to compute the XOR-operation twice as fast as with a classical algorithm makes it obvious that the parity of the blackbox X can be calculated by N/2 oracle calls. The procedure proposed by Beals *et al.* calculates the parity of each of the $N/2$ pairs $(x_0, x_1), (x_2, x_3), \ldots, (x_{N-2}, x_{N-1})$ separately [BBC+01]. This can be done with $N/2$ oracle calls by the procedure shown above. Thus one obtains a list of $N/2$ binary values representing the parity of each of these pairs. Now one calculates the parity of this list to obtain the parity of the blackbox X. The final step can be done without any further oracle calls. Thus, one has to perform $N/2$ measurements to solve this problem.

The algorithm we present next solves this problem with $N/2$ oracle calls and a single n-qubit measurement.

A.2 Parity Algorithm proposed by Farhi

The QA proposed by Farhi *et al.* [FGGS98] starts with the initial n-qubit state $|\psi_0\rangle = |\underline{0}\rangle$ and creates a superposition over all $N = 2^n$ states by applying a Hadamard gate to each of the n qubits:

$$|\psi_1\rangle = \boldsymbol{H}^{\otimes}|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\underline{i}\rangle.$$

Now Farhi *et al.* define the unitary operation $\boldsymbol{V}$:[2]

$$\begin{aligned} \boldsymbol{V}|\underline{i}\rangle &= |\underline{i+1}\rangle \text{ for } i = 0, \ldots, N/2-2 \quad , \qquad \boldsymbol{V}|\underline{N/2-1}\rangle = |\underline{0}\rangle, \\ \boldsymbol{V}|\underline{i}\rangle &= |\underline{i+1}\rangle \text{ for } i = N/2, \ldots, N-2 \quad , \qquad \boldsymbol{V}|\underline{N-1}\rangle = |\underline{N/2}\rangle. \end{aligned}$$

Using this gate the next state $|\psi_2\rangle$ is obtained by $N/2$ repetitions of the gate sequence $\boldsymbol{V} \cdot \boldsymbol{O}$, here $\boldsymbol{O}$ denotes the oracle gate as defined in Eq. (3.4.1). One gets:

$$\begin{aligned} |\psi_2\rangle &= \underbrace{\boldsymbol{VO} \cdot \boldsymbol{VO} \cdots \boldsymbol{VO}}_{N/2\text{-times}} |\psi_1\rangle \\ &= \frac{1}{\sqrt{N}} \left[(-1)^{x_0 \oplus x_1 \oplus \cdots \oplus x_{N/2-1}} \left(\sum_{i=0}^{N/2-1} |\underline{i}\rangle \right) + (-1)^{x_{N/2} \oplus x_{N/2+1} \oplus \cdots \oplus x_{N-1}} \left(\sum_{i=N/2}^{N-1} |\underline{i}\rangle \right) \right] \\ &= \frac{1}{\sqrt{N}} (-1)^{x_0 \oplus x_1 \oplus \cdots \oplus x_{N/2-1}} \left[\sum_{i=0}^{N/2-1} |\underline{i}\rangle + (-1)^{x_0 \oplus x_1 \oplus \cdots \oplus x_{N-1}} \left(\sum_{i=N/2}^{N-1} |\underline{i}\rangle \right) \right]. \end{aligned}$$

As global phases cannot be measured the measurement result depends on the local phase $(-1)^{x_0 \oplus x_1 \oplus \cdots \oplus x_{N-1}} = (-1)^{f(X)}$. Applying a Hadamard gate on each of the n qubits leaves us with two possibilities: for $f(X) = 0$ the state $|\psi_2\rangle$ is mapped to $|\underline{0}\rangle$, for $f(x) = 1$ the state $|\psi_2\rangle$ is mapped to a superposition where the state $|\underline{0}\rangle$ vanishes. Thus, an n-qubit measurement reveals the parity $f(X)$ of the blackbox X.

[2] The authors do not mention how efficient $\boldsymbol{V}$ can be decomposed into elementary quantum gates. A straightforward decomposition of $\boldsymbol{V}$ without introducing any ancillary qubits needs $\mathcal{O}(n^4)$ one- and two-qubit operations.

B Notes on the NMR Experiment

According to the definition of the oracle gate in Eq. (4.4) the blackbox $X = (0,1,0,1)$, abbreviated by **0101**, has the form:

$$\mathbf{0101} \quad \sim \quad \boldsymbol{O} = \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix}$$

As global phase shifts are not measurable this oracle gate can also be used to represent the blackbox $X = (1,0,1,0)$. Thus, one only has to implement oracles representing the blackboxes $X = (0,0,0,0)$, $X = (0,0,1,1)$, $X = (0,1,0,1)$, $X = (0,1,1,0)$, $X = (0,0,0,1)$, $X = (0,0,1,0)$, $X = (0,1,0,0)$ and $X = (1,0,0,0)$:

$$\mathbf{0011} \quad \sim \quad \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix} \sim \left(e^{-\imath\frac{\pi}{2}\boldsymbol{\sigma}_x} \otimes \mathbf{1}\right) \cdot \left(e^{-\imath\frac{\pi}{2}\boldsymbol{\sigma}_y} \otimes \mathbf{1}\right) \equiv (\pi)_y^X - (\pi)_x^X \tag{B.1}$$

$$\mathbf{0101} \quad \sim \quad \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix} \sim \left(\mathbf{1} \otimes e^{-\imath\frac{\pi}{2}\boldsymbol{\sigma}_x}\right) \cdot \left(\mathbf{1} \otimes e^{-\imath\frac{\pi}{2}\boldsymbol{\sigma}_y}\right) \equiv (\pi)_y^A - (\pi)_x^A$$

$$\mathbf{0110} \quad \sim \quad \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & -1 & \\ & & & 1 \end{pmatrix} \sim e^{-\imath\frac{\pi}{2}\boldsymbol{\sigma}_z \otimes \boldsymbol{\sigma}_z} \equiv (2\pi)^{AX}$$

$$\mathbf{0001} \quad \sim \quad \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix} \sim \left(e^{\imath\frac{\pi}{4}\boldsymbol{\sigma}_z} \otimes \mathbf{1}\right) \left(\mathbf{1} \otimes e^{\imath\frac{\pi}{4}\boldsymbol{\sigma}_z}\right) \left(e^{-\imath\frac{\pi}{4}\boldsymbol{\sigma}_z \otimes \boldsymbol{\sigma}_z}\right)$$

$$\equiv \quad (\pi)^{AX} - \left(\frac{\pi}{2}\right)_{-z}^X - \left(\frac{\pi}{2}\right)_{-z}^A$$

$$\mathbf{0010} \quad \sim \quad \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & 1 \end{pmatrix} \sim \left(e^{-\imath\frac{\pi}{4}\boldsymbol{\sigma}_z} \otimes \mathbf{1}\right) \left(\mathbf{1} \otimes e^{\imath\frac{\pi}{4}\boldsymbol{\sigma}_z}\right) \left(e^{-\imath\frac{\pi}{4}\boldsymbol{\sigma}_z \otimes \boldsymbol{\sigma}_z}\right)$$

$$\equiv \quad (\pi)^{AX} - \left(\frac{\pi}{2}\right)_{z}^X - \left(\frac{\pi}{2}\right)_{-z}^A$$

$$
\begin{aligned}
\mathbf{0100} &\sim \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \sim \left(e^{\imath\frac{\pi}{4}\boldsymbol{\sigma}_z}\otimes\mathbf{1}\right)\left(\mathbf{1}\otimes e^{-\imath\frac{\pi}{4}\boldsymbol{\sigma}_z}\right)\left(e^{-\imath\frac{\pi}{4}\boldsymbol{\sigma}_z\otimes\boldsymbol{\sigma}_z}\right) \\
&\equiv (\pi)^{AX} - \left(\frac{\pi}{2}\right)^{X}_{-z} - \left(\frac{\pi}{2}\right)^{A}_{z} \\
\mathbf{1000} &\sim \begin{pmatrix} -1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \sim \left(e^{-\imath\frac{\pi}{4}\boldsymbol{\sigma}_z}\otimes\mathbf{1}\right)\left(\mathbf{1}\otimes e^{-\imath\frac{\pi}{4}\boldsymbol{\sigma}_z}\right)\left(e^{-\imath\frac{\pi}{4}\boldsymbol{\sigma}_z\otimes\boldsymbol{\sigma}_z}\right) \\
&\equiv (\pi)^{AX} - \left(\frac{\pi}{2}\right)^{X}_{z} - \left(\frac{\pi}{2}\right)^{A}_{z}
\end{aligned}
$$

Now all oracle gates are decomposed into NMR pulse sequences. As the gates $\left(\frac{\pi}{2}\right)_z$, $\left(\frac{\pi}{2}\right)_{-z}$, $(\pi)_z$ and $(\pi)_{-z}$ cannot be implemented directly on an NMR-QC one can use the following decompositions to realize these gates:

$$
\begin{aligned}
(\pi)_z &= (\pi)_x - (\pi)_y \\
(\pi)_{-z} &= (\pi)_{-x} - (\pi)_y \\
\left(\frac{\pi}{2}\right)_z &= \left(\frac{\pi}{2}\right)_x - \left(\frac{\pi}{2}\right)_y - \left(\frac{\pi}{2}\right)_{-x} \\
\left(\frac{\pi}{2}\right)_{-z} &= \left(\frac{\pi}{2}\right)_x - \left(\frac{\pi}{2}\right)_{-y} - \left(\frac{\pi}{2}\right)_{-x}
\end{aligned}
\tag{B.2}
$$

B.1 Pulse Sequences

$$
\begin{aligned}
\boldsymbol{CNOT}_{XA} &\equiv \left(\tfrac{\pi}{2}\right)^{(X)}_{y} - (\pi)^{(AX)} - \left(\tfrac{\pi}{2}\right)^{(X)}_{-y} - \left(\tfrac{\pi}{2}\right)^{(X)}_{x} - \left(\tfrac{\pi}{2}\right)^{(A)}_{-y} - \left(\tfrac{\pi}{2}\right)^{(A)}_{-x} - \left(\tfrac{\pi}{2}\right)^{(A)}_{y} \\
\boldsymbol{CNOT}_{AX} &\equiv \left(\tfrac{\pi}{2}\right)^{(A)}_{y} - (\pi)^{(AX)} - \left(\tfrac{\pi}{2}\right)^{(A)}_{-y} - \left(\tfrac{\pi}{2}\right)^{(A)}_{x} - \left(\tfrac{\pi}{2}\right)^{(X)}_{-y} - \left(\tfrac{\pi}{2}\right)^{(X)}_{-x} - \left(\tfrac{\pi}{2}\right)^{(X)}_{y} \\
\boldsymbol{R}_y(\frac{\pi}{2}) &\equiv \left(\tfrac{\pi}{2}\right)_y \\
\boldsymbol{R}_y(-\frac{\pi}{2}) &\equiv \left(\tfrac{\pi}{2}\right)_{-y} \\
\boldsymbol{\sigma_x} &\equiv (\pi)_x \\
\boldsymbol{Refocus} &\equiv \left(\tfrac{\pi}{2}\right)^{(AX)} - (\pi)^{(X)}_{x} - \left(\tfrac{\pi}{2}\right)^{(AX)} - (\pi)^{(X)}_{x} \equiv \mathbf{1} \\
\mathbf{0000} &\equiv \mathbf{1} \\
\mathbf{0011} &\equiv (\pi)^{X}_{y} - (\pi)^{X}_{x} \\
\mathbf{0101} &\equiv (\pi)^{A}_{y} - (\pi)^{A}_{x} \\
\mathbf{0110} &\equiv (2\pi)^{AX} \\
\mathbf{0001} &\equiv (\pi)^{AX} - \left(\tfrac{\pi}{2}\right)^{X}_{x} - \left(\tfrac{\pi}{2}\right)^{X}_{-y} - \left(\tfrac{\pi}{2}\right)^{X}_{-x} - \left(\tfrac{\pi}{2}\right)^{A}_{x} - \left(\tfrac{\pi}{2}\right)^{A}_{-y} - \left(\tfrac{\pi}{2}\right)^{A}_{-x} \\
\mathbf{0010} &\equiv (\pi)^{AX} - \left(\tfrac{\pi}{2}\right)^{X}_{x} - \left(\tfrac{\pi}{2}\right)^{X}_{y} - \left(\tfrac{\pi}{2}\right)^{X}_{-x} - \left(\tfrac{\pi}{2}\right)^{A}_{x} - \left(\tfrac{\pi}{2}\right)^{A}_{-y} - \left(\tfrac{\pi}{2}\right)^{A}_{-x}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{0100} &\equiv (\pi)^{AX} - (\tfrac{\pi}{2})^{X}_{x} - (\tfrac{\pi}{2})^{X}_{-y} - (\tfrac{\pi}{2})^{X}_{-x} - (\tfrac{\pi}{2})^{A}_{x} - (\tfrac{\pi}{2})^{A}_{y} - (\tfrac{\pi}{2})^{A}_{-x} \\
\mathbf{1000} &\equiv (\pi)^{AX} - (\tfrac{\pi}{2})^{X}_{x} - (\tfrac{\pi}{2})^{X}_{y} - (\tfrac{\pi}{2})^{X}_{-x} - (\tfrac{\pi}{2})^{A}_{x} - (\tfrac{\pi}{2})^{A}_{y} - (\tfrac{\pi}{2})^{A}_{-x} \\
\boldsymbol{Readout} &\equiv (\tfrac{\pi}{2})^{(X)}_{y}
\end{aligned}
$$

B.2 Parity Algorithm for a Single 2-Qubit Quantum System

The first Hadamard gate of the algorithm in the upper part of Fig. 5.7 can be replaced by $\boldsymbol{R}_y(\pi/2) = e^{-\imath\pi/4\boldsymbol{\sigma}_y} = \frac{1}{\sqrt{2}}(\mathbf{1} - \imath\boldsymbol{\sigma}_y)$, the final Hadamard gate can be replaced by $\boldsymbol{R}_y(-\pi/2) = \frac{1}{\sqrt{2}}(\mathbf{1} + \imath\boldsymbol{\sigma}_y)$.

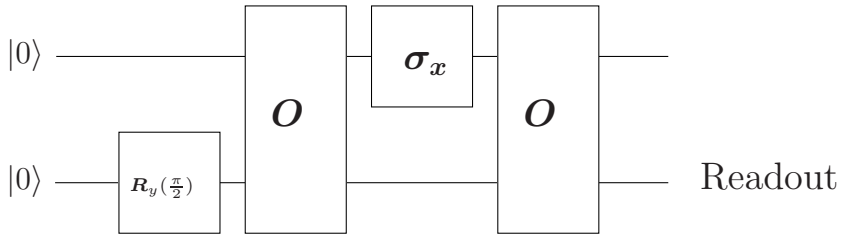

Figure B.1: Modified version of the parity QC for a single two-qubit quantum system. The last pseudo-Hadamard gate cancels out the $\boldsymbol{R}_y(\frac{\pi}{2})$-readout pulse.

B.3 The Experiment

- Create one of the three initial states as shown in equation (2.27):
  $[n_{00}, n_{01}, n_{10}, n_{11}]$
  $[n_{00}, n_{11}, n_{01}, n_{10}]$: $(\boldsymbol{CNOT}_{AX})(\boldsymbol{CNOT}_{XA})$
  $[n_{00}, n_{10}, n_{11}, n_{01}]$: $(\boldsymbol{CNOT}_{XA})(\boldsymbol{CNOT}_{AX})$
- Now perform the pseudo-Hadamard gate $\boldsymbol{R}_y(\pi/2)$ to spin X.
- Perform one of the oracle gates.
- Apply the $\boldsymbol{NOT}$-operation $\boldsymbol{\sigma}_x$ to spin A.
- Perform the same oracle gate again.
- Readout of spin X.
- Repeat this procedure two times for all of the different initial states from above and sum up the spectra.

C Evolution of the Parity QC

In this chapter we will examine how the GP system evolves the two-qubit parity QC presented in Fig. 5.7 (a). The parameters of the GP system are shown in Tab. C.1.

Population size	10
No. of generations	1000
Tournament size	4
Crossover probability	0.05
Creation probability	0.05
Mutation probability	0.90
Swap mutation probability	0.30×0.9
Grow mutation probability	0.30×0.9
Shrink mutation probability	0.20×0.9
Shrink2 mutation probability	0.20×0.9
No. of rotation angles	128
Min. no. of gates	5
Max. no. of gates	100
Max. no. of oracle gates	16
Gate set	$\boldsymbol{H}$, $\boldsymbol{CNOT}$, $\boldsymbol{R}_x(2 \cdot \theta_l)$, $\boldsymbol{R}_y(2 \cdot \theta_l)$, $\boldsymbol{O}$

Table C.1: Parameters of the GP system that evolved the QC depicted in Fig. 5.7 (a). The rotation angle θ_l is specified by the integer $l \in \{0, 1, ..., 127\}$ via $\theta_l = -\pi + (l + 1) \cdot \frac{2\pi}{128}$.

In order to illustrate the evolutionary process we have chosen a population of only ten individuals. The individuals of the initial population and the corresponding fitness values are shown in Tab. C.2. It follows that individual 3 is the fittest one. The initial population is created completely at random.

The flowchart in Fig. C.1 illustrates the different stages of the evolutionary process in the GP system: A new generation is created from the initial one by tournament selection. Due to the parameter in Tab. C.1 each tournament set consists of four individuals. These individuals are chosen at random. According to the mutation and crossover probabilities in Tab. C.1 the GP system decides whether a mutation or crossover occurs. The new individuals that are generated by these genetic operators are inserted into the new population. The best individual of the current population is copied unaltered to the new population (reproduction). This procedure is repeated until the new population consists of ten individuals. Then the fitness values are calculated. When the fitness of one of these new individuals meets the termination condition the whole process is stopped and the GP system designates this individual. Otherwise, a new population is created from the current one by mutations, crossovers and reproduction. This process is repeated until an individual meets the termination criterion.

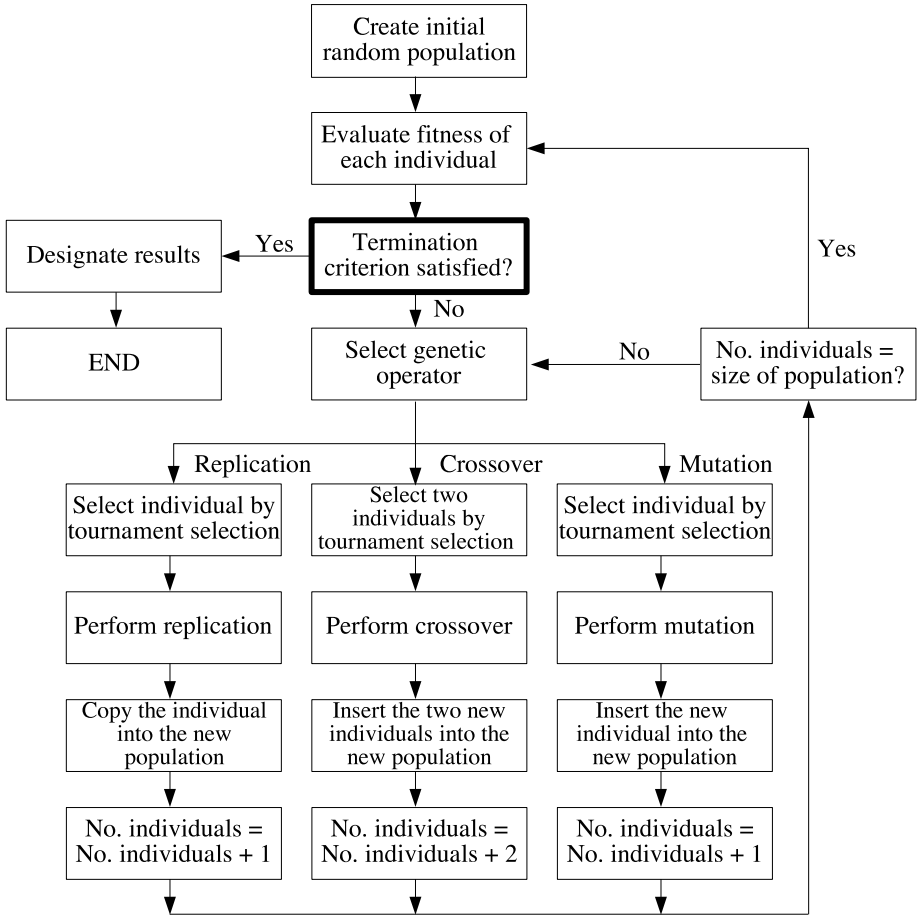

Figure C.1: This flowchart illustrates the different stages of the evolutionary process in the GP system. The size of the population does not change during an evolutionary run.

Tab. C.3 presents some data of the best individual of each generation found by the GP system. The corresponding gate sequence is listed in Tab. C.4. Only those generations are shown whose best individual has a higher fitness than the ones of former generations. The gray scales in Tab. C.4 serve to illustrate the relationship between the best individuals of each generation.

The statistics illustrated in the graphs of Fig. C.2 to Fig. C.5 indicate that this example run is a typical run of the GP system. In order to generate these statistics we performed 100 independent runs. The results are visualized via quartiles.[1]

The graph in Fig. C.2 shows the length of the best individual found in each genera-

[1] In order to find a quartile all values of a random variable are arranged from the lowest to the highest value. This sorted set is divided into four equal parts so that each part represents 1/4th of all values. The first quartile cuts off the lowest 1/4th of all values, the second quartile (median) cuts of the lower half and the third quartile cuts of the lower 3/4th of all values.

tion. Fig. C.3 presents the corresponding values of the fitness variable `worst_error`, and Fig. C.4 shows the number of oracle gates. Also, each figure displays the values of the example run.

The results that were presented in this chapter show that the number of oracle gates changes during an evolutionary run of the GP system. Also, the GP system changes the measurement qubits as can be seen in Fig. C.5. Hence, it is not necessary to fix the measurement qubits and the number of oracles in the beginning of each run.

Additional Remarks: Finally we have to make some remarks on the values of `avg_error` and `worst_error` shown in Tab. C.2, Tab. C.3 and Fig. C.3: According to the definition of the fitness function in Sec. 4.4 we only calculate `avg_error` and `worst_error` when the value of `hits` is equal to zero. Also, according to this description a value of `worst_error` that is bigger than 0.5 makes no sense. Nevertheless, the values presented in Tab. C.2, Tab. C.3 and Fig. C.3 are calculated even though `hits` is not zero. In such a case the values of `worst_error` and `avg_error` do not represent error probabilities. Anyhow, these values can be used in the evaluation of the fitness. The values of `avg_error` presented in Tab. C.2 and Tab. C.3 represent the sum of all errors. This makes no difference for the fitness function and can be computed faster.

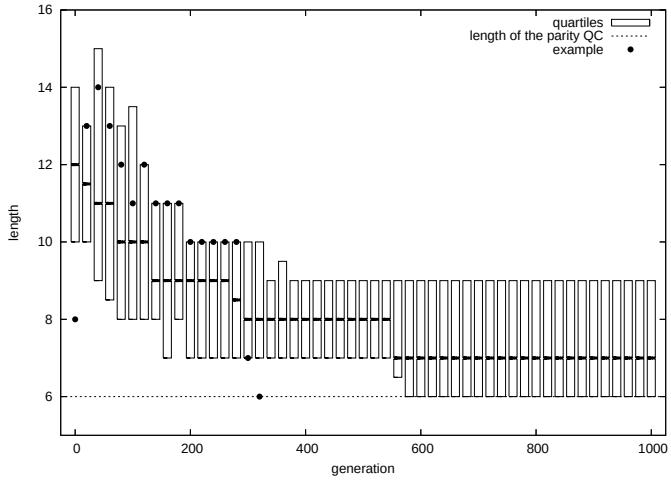

Figure C.2: This graph shows the length distribution (first, second and third quartiles) of the best individual in generation n for 100 independent runs of the GP system. Horizontal black bars denote the median. The lower, upper end of the boxes indicates the first, third quartile, respectively. Black dots denote the corresponding values of the example run.

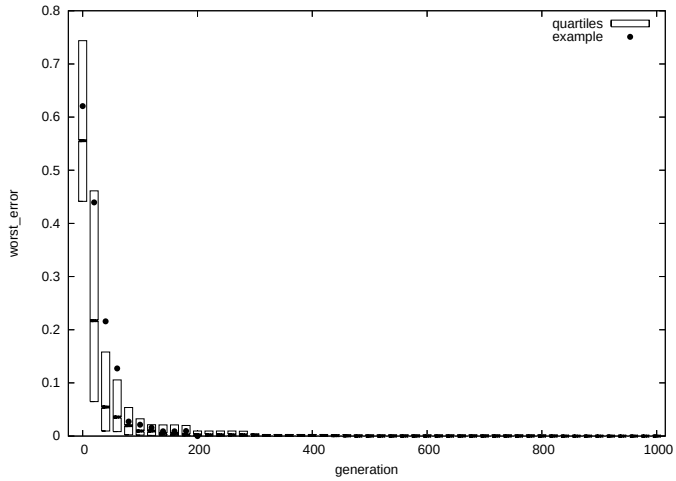

Figure C.3: This graph shows the `worst_error` distribution (first, second and third quartiles) of the best individual in generation n for 100 independent runs of the GP system. Horizontal black bars denote the median. The lower, upper end of the boxes indicates the first, third quartile, respectively. Black dots denote the corresponding values of the example run.

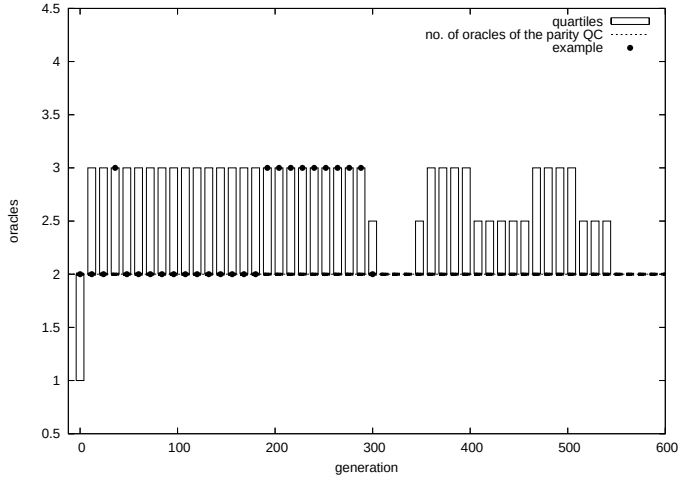

Figure C.4: This graph shows the distribution (first, second and third quartiles) of the number of oracle gates of the best individual in generation n for 100 independent runs of the GP system. Horizontal black bars denote the median. The lower, upper end of the boxes indicates the first, third quartile, respectively. Black dots denote the corresponding values of the example run.

Individual 0	Individual 1	Individual 2	Individual 3
NOOP CNOT[1 0] Ry[0 117] Ry[0 99] END	CNOT[1 0] Ry[0 61] Ry[0 13] CNOT[1 0] Rx[0 84] END	CNOT[1 0] CNOT[0 1] CNOT[1 0] Rx[1 31] HAD[1] HAD[1] END	Rx[0 127] ORACLE Rx[0 21] CNOT[1 0] HAD[0] ORACLE Rx[0 2] END
Fitness invalid (no oracles)	**Fitness** invalid (no oracles)	**Fitness** invalid (no oracles)	**Fitness** hits: 8 a_err: 14.07 w_err: 0.62
Individual 4	**Individual 5**	**Individual 6**	**Individual 7**
ORACLE ORACLE CNOT[1 0] NOOP CNOT[0 1] NOOP Rx[0 1] CNOT[1 0] END	HAD[1] NOOP HAD[0] CNOT[0 1] HAD[0] Ry[1 52] Ry[0 60] NOOP Rx[1 62] END	ORACLE NOOP NOOP ORACLE Rx[1 112] ORACLE Ry[1 112] CNOT[1 0] NOOP Rx[1 3] END	CNOT[0 1] Ry[0 114] HAD[0] ORACLE NOOP CNOT[1 0] CNOT[1 0] Ry[0 21] ORACLE CNOT[0 1] NOOP[0 1] END
Fitness hits: 16 a_err: 16 w_err: 1	**Fitness** invalid (no oracles)	**Fitness** hits: 16 a_err: 16 w_err: 0.5044	**Fitness** hits: 16 a_err: 14.07 w_err: 0.8865
Individual 8	**Individual 9**		
Ry[0 89] ORACLE CNOT[1 0] NOOP Ry[0 108] HAD[0] ORACLE CNOT[1 0] ORACLE Ry[1 80] Rx[1 1] CNOT[1 0] END	NOOP Rx[0 26] Rx[1 110] NOOP Ry[1 120] Ry[1 25] ORACLE Ry[1 39] Ry[0 13] Ry[1 66] ORACLE HAD[1] HAD[1] END		
Fitness hits: 16 a_err: 14.71 w_err: 0.5363	**Fitness** hits: 8 a_err: 15.34 w_err: 0.5148		

Table C.2: This table lists the 10 individuals of our starting population. These individuals are generated at random. The gate sequence is to be read from the bottom to the top. `H[1]` denotes a Hadamard gate $\boldsymbol{H}$ that is applied to qubit 1. `CNOT[0 1]` denotes a $\boldsymbol{CNOT}$ gate with qubit 0 the control and qubit 1 the target qubit. `Ry[0 117]` denotes that the one-qubit gate $\boldsymbol{R}_y(2 \cdot \theta_l)$ is applied to qubit 0. The angle θ_l is defined by: $\theta_l = -\pi + (l+1) \cdot \frac{2\pi}{128}$. One gets: $\theta_{117} = 54\pi/64$. `NOOP` denotes an identity operation and does nothing. `END` denotes the end of the gate sequence. The fitness values `hits`, `avg_error` and `worst_error` are shown at the end of each gate sequence.

	Lgth	#Gates			#Ind	#Gen. Op.		Fitness			
gen	**bst**	**O**	**N**	**P**	**bst**	**mut**	**crs**	**hits**	**err**	**werr**	**msr**
0	8	2	0	1	3	0	0	8	14.07	0.621	0
1	11	2	2	4	7	1	0	8	9.348	0.916	0
3	10	2	2	3	7	2	0	8	9.348	0.916	0
6	9	1	2	3	7	3	0	8	9.348	0.916	0
7	10	1	2	3	7	4	0	8	9.348	0.846	0
9	11	2	2	4	7	5	1	8	8	1	0
15	10	2	1	3	7	11	1	8	8	0.778	0
16	9	2	1	2	7	12	1	8	8	0.778	0
19	13	3	1	3	7	15	1	8	5.946	0.44	0 1
22	13	4	1	3	7	16	1	8	5.916	0.443	0 1
23	16	3	1	5	7	16	2	8	5.916	0.443	0 1
24	15	2	1	5	7	17	2	8	5.916	0.443	0 1
25	14	2	1	5	7	18	2	8	5.639	0.409	0 1
27	15	3	1	4	7	20	2	0	7.002	0.162	0 1
29	14	3	1	3	7	21	2	0	7.002	0.162	0 1
30	16	3	1	4	7	22	2	0	6.082	0.216	0 1
33	15	3	1	3	7	23	2	0	6.082	0.216	0 1
34	14	3	1	2	7	24	2	0	6.082	0.216	0 1
41	13	2	1	2	7	25	2	0	6.082	0.216	0 1
44	15	2	2	2	7	27	2	0	1.657	0.127	0 1
51	14	2	2	2	7	28	2	0	1.657	0.127	0 1
52	13	2	1	2	7	29	2	0	1.657	0.127	0 1
76	12	2	0	2	7	30	2	0	1.657	0.127	0 1
79	12	2	0	3	7	30	3	0	0.836	0.027	0 1
84	11	2	0	2	7	31	3	0	0.836	0.027	0 1
87	11	2	1	2	7	32	3	0	0.836	0.021	0 1
106	11	2	0	2	7	33	3	0	0.836	0.021	0 1
109	12	2	0	2	7	34	3	0	0.836	0.016	0 1
128	12	2	0	2	7	35	3	0	0.154	0.01	1
133	11	2	0	2	7	36	3	0	0.154	0.009	0 1
162	11	2	0	2	7	38	3	0	0.078	0.01	0 1
189	10	2	0	1	7	44	4	0	0.078	0.01	0 1
191	11	3	0	1	7	45	4	0	0.002	0	0 1
193	10	3	0	0	7	47	4	0	0.002	0	0 1
289	8	2	0	0	7	50	4	0	0	0	1
295	7	2	0	0	7	51	4	0	0	0	1
307	6	2	0	0	7	52	4	0	0	0	1

Table C.3: This table is to be read from the top to the bottom and illustrates the evolutionary process. The leftmost column shows the current generation. For each generation some values of the best individual are shown in the corresponding row. **Lgth** denotes the length this individual. **O**, **N** and **P** denote the number of its `ORACLE`, `NOOP` and `CNOT` gates, respectively. **bst** is the ID of the best individual. **mut** and **crs** denote the total number of mutation and crossover operations, respectively, that were applied to this individual. **hits**, **err** and **werr** denote the fitness values `hits`, `avg_error` and `worst_error`. **msr** denotes the qubits that are to be measured.

Gen	Gate Sequence of Individual 7																			
0		C[0 1]	O	R2	C[1 0]		C[1 0]		O	H[0]	R4	C[0 1]								
1		C[0 1]	O	R2	C[1 0]		C[1 0]		O	H[0]		C[0 1]								
3		C[0 1]	O	R2	C[1 0]				O	H[0]		C[0 1]								
6		C[0 1]		R2	C[1 0]				O	H[0]		C[0 1]								
7	R1	C[0 1]		R2	C[1 0]				O	H[0]		C[0 1]								
9		C[0 1]	O	H[0]	C[1 0]		C[1 0]		O	H[0]		C[0 1]								
15		C[0 1]		R2	C[1 0]	H[0]			O	H[0]	O	C[0 1]								
16		C[0 1]		R2	C[1 0]	H[0]			O	H[0]	O									
19		C[0 1]		H[0]	C[1 0]	H[0]	C[1 0]		O	H[0]	O	R5						R11		O
22		C[0 1]		H[0]	C[1 0]	H[0]	C[1 0]		O	H[0]	O					O		R11		O
23		C[0 1]		H[0]	C[1 0]	H[0]	C[1 0]		O	H[0]	H[0]	C[0 1]		H[0]	C[1 0]	O		R11		O
24		C[0 1]		H[0]	C[1 0]	H[0]	C[1 0]		O	H[0]	H[0]	C[0 1]		H[0]	C[1 0]			R11		O
25		C[0 1]		H[0]	C[1 0]	H[0]	C[1 0]		O	H[0]		C[0 1]		H[0]	C[1 0]			R11		O
27		C[0 1]		H[0]	C[1 0]		C[1 0]		O	H[0]	H[0]	C[0 1]		H[0]	R8	O		R11		O
29	H[0]	C[0 1]					C[1 0]		O	H[0]	H[0]	C[0 1]		H[0]	R8	O		R11		O
30		C[0 1]		H[0]	C[0 1]		C[1 0]		O	H[0]	H[0]	C[0 1]		H[0]	R8	O		R11	R6	O
33		C[0 1]		H[0]	C[0 1]		C[1 0]		O	H[0]	H[0]			H[0]	R8	O		R11	R6	O
34	H[0]	C[0 1]					C[1 0]		O	H[0]	H[0]			H[0]	R8	O		R11	R6	O
41	H[0]	C[0 1]					C[1 0]		O	H[0]	H[0]			H[0]	R8	O		R11	R6	
44	H[0]	C[0 1]					C[1 0]		O	H[0]	H[0]			H[0]	R8	O	R9	R11	R6	
51	H[0]	C[0 1]					C[1 0]		O	H[0]	H[0]			H[0]	R8	O	R10		R6	
79	H[0]	C[0 1]					C[1 0]		O			C[0 1]	R6	H[0]	R8	O	R10		R6	
84	H[0]	C[0 1]					C[1 0]		O				R6	H[0]	R8	O	R10		R6	
87	H[0]	C[0 1]					C[1 0]		O				R6	H[0]	R8	O	R10			
106	H[0]	C[0 1]					C[1 0]		O				R6	H[0]	R8	O	R10		R12	
109	H[0]	C[0 1]					C[1 0]	R3	O				R6	H[0]	R8	O	R10		R12	
128	H[0]	C[0 1]					C[1 0]	R3	O				R6	H[0]	R8	O	H[1]		R12	
133	H[0]	C[0 1]					C[1 0]		O				R6	H[0]	R8	O	H[1]		R12	
162	H[0]	C[0 1]					C[1 0]		O				R6	H[0]	R8	O	H[1]		R13	
189		C[0 1]		H[1]					O				R6	H[0]	R8	O	H[1]		R13	
191		C[0 1]		H[1]					O				R6	H[0]	R8	O	H[1]	O	R13	
193				H[1]					O				R6	H[0]	R8	O	H[1]	O	R13	
289				H[1]					O				R7		R8	O	H[1]		R13	
295				H[1]					O				R7		R8	O	H[1]			
307				H[1]					O					R14		O	H[1]			

Table C.4: This table illustrates the evolutionary process. The leftmost column shows the current generation. The gate sequence of individual 7 (best individual, except in generation 0) is shown in the corresponding row. This gate sequence is to be read from left to right. `NOOP` and `END` operations are not shown. The fitness and other values of this gate sequence are presented in Tab. C.3. Gray scales are intended to guide the eye. H:`HAD`, C:`CNOT`, O:`ORACLE`, R1:`Ry[0 37]`, R2:`Ry[0 21]`, R3:`Rx[0 120]`, R4:`Ry[0 114]`, R5:`Rx[1 1]`, R6:`Ry[0 58]`, R7:`Ry[0 86]`, R8:`Ry[0 8]`, R9:`Ry[1 83]`, R10:`Ry[1 12]`, R11:`Ry[1 120]`, R12:`Rx[0 58]`, R13:`Ry[0 33]`, R14:`Ry[0 31]`.

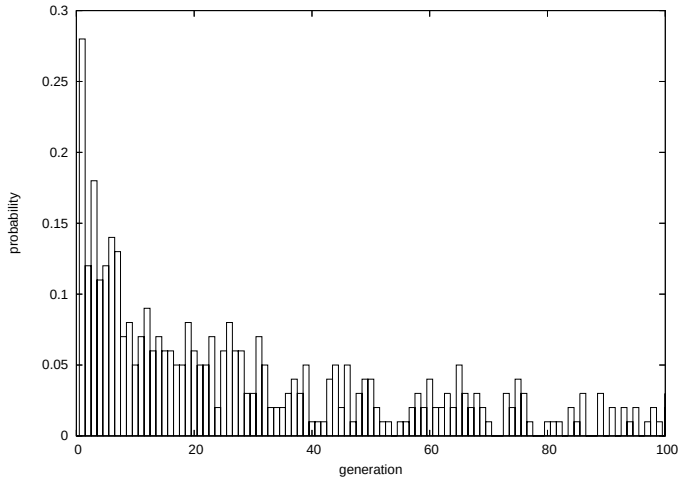

Figure C.5: This graphs shows the probability that the GP system changes the measurement qubits in the nth generation.

About the Author

The author studied physics from 1995 to 2001 at the University of Konstanz, Germany. There, he received the diploma in physics (Dipl.-Phys.) in 2001. The study focused on computational and theoretical physics.

The diploma thesis dealt with the numerical investigation of a two-dimensional spinfluid. These studies were performed at the chair of Statistical Physics at the University of Konstanz under the supervision of Prof. Dr. Peter Nielaba.

From 2001 to 2004 the author was a scholarship holder within the Ph. D. program *Materials and Concepts of Quantum Information Processing* at the University of Dortmund, Germany. There, under the supervision of Prof. Dr. Wolfgang Banzhaf and Prof. Dr. Dieter Suter, he accomplished the investigations that are presented in this thesis.

Since then the author works as a programmer and system administrator at the chair of Bioinformatics at the University of Konstanz. There, his interests focus mainly on the protein folding problem that he tackles by using so called simple exact models.

In his spare time the author either observes recent developments in the field of quantum computation or studies epistemological problems of science.

Publications

Ralf Stadelhofer, Dieter Suter and Wolfgang Banzhaf, *Quantum and Classical Parallelism in Parity Algorithms for Ensemble Quantum Computers*, Phys. Rev. A **71**, 032345 (2005)

R. Stadelhofer, D. Suter, and W. Banzhaf, *Evolving Blackbox Quantum Algorithms using Genetic Programming*, to be published in Computer-Aided Design.

Bibliography

[AC03] Arvind and D. Collins. Scaling Issues in Ensemble Implementations of the Deutsch-Jozsa Algorithm. *Phys. Rev. A*, 68:052301, 2003. 85

[AdW01] A. Ambainis and R. de Wolf. Average-Case Quantum Query Complexity. *J. Phys. A: Math. Gen.*, 34:6741, 2001. 38, 59

[AMS04] I.J.R. Aitchison, D.A. MacManus, and T.M. Snyder. Understanding Heisenberg's Magical Paper of July 1925: A new Look at the Calculation Details. *Am. J. Phys.*, 72:1370, 2004. 5

[ASW02] A. Acin, V. Scarani, and M.M. Wolf. Bell's Inequalities and Distibillity in N-quantum-bit Systems. *Phys. Rev. A*, 66:042323, 2002. 15

[Aud05] J. Audretsch. *Verschränkte Systeme*. WILEY-VCH Verlag, Weinheim, 2005. 16

[Bal98] L.E. Ballentine. *Quantum Mechanics: A Modern Development*. World Scientific Publishing Company, 1998. 3

[BB92] A. Berthiaume and G. Brassard. The Quantum Challenge to Structural Complexity Theory, 1992. 34

[BB94] A. Berthiaume and G. Brassard. Oracle Quantum Computing. *J. Mod. Opt.*, 41:2521, 1994. 34

[BBBV97] C.H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM J. Comput.*, 26(5):1510, 1997. 1

[BBC+01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum Lower Bounds by Polynomials. *JACM*, 48:778, 2001. 1, 25, 36, 37, 38, 42, 65, 81, 83, 89, 91, 93, 94

[BBS00] H. Barnum, H.J. Bernstein, and L. Spector. Quantum Circuits for OR and AND of ORs. *J. Phys. A: Math. Gen.*, 33:8047, 2000. 2, 51

[BdW02] H. Burhman and R. de Wolf. Complexity Measures and Decision Tree Complexity: A Survey. *Theoretical Computer Science*, 288:21, 2002. 25, 35, 36, 53

[Bel87] J.S. Bell. *Speakable and Unspeakable in Quantum Mechanics*. Cambridge University, Cambridge, 1987. 1, 3, 10, 13, 14

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=?NP Question. *SIAM J. Comput.*, 4:431, 1975. 34

[BH97] G. Brassard and P. Høyer. An Exact Quantum Polynomial-Time Algorithm for Simon's Problem. In IEEE, editor, *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems (ISTCS)*, page 12, Silver Spring, MD, USA, Jun. 1997. IEEE Computer Society Press. 78

[BHJ25] M. Born, W. Heisenberg, and P. Jordan. Zur Quantenmechanik II. *Z. Phys.*, 35:557, 1925. 5

[BJ25] M. Born and J. Jordan. Zur Quantenmechanik. *Z. Phys.*, 34:858, 1925. 5

[BNKF98] W. Banzhaf, P. Nordin, R.E. Keller, and F.D. Francone. *Genetic Programming: An Introduction.* Morgan Kaufmann Publishers, 1998. 51, 53

[Boh13] N. Bohr. On the Constitution of Atoms and Molecules. *Phil. Mag.*, 26:1, 1913. 4

[Boh23] N. Bohr. Die Grundpostulate der Quantenmechanik. *Z. Phys.*, 13:117, 1923. 4

[Boh49] N. Bohr. Discussion with Einstein on Epistomological Problems in Atomic Physics. In P.A. Schlipp, editor, *Albert Einstein: Philosopher-Scientist*, page 200. The Library of Living Philosophers, Evanston, 1949. 1, 10

[Bor26] M. Born. Zur Quantenmechanik der Stoßvorgänge. *Z. Phys.*, 37:863, 1926. 7

[BS02] H.G. Beyer and H.P. Schwefel. Evolution strategies – A Comprehensive Introduction. *Natural Computing*, 1(1):3, 2002. 53

[BV97] E. Bernstein and U. Vazirani. Quantum Complexity Theory. *SIAM J. Comput.*, 26:1411, 1997. 1, 29, 31, 33

[BZ05] C. Brukner and A. Zeilinger. Quantum Physics as a Science of Information. In A. Elitzur, S. Dolev, and N. Kolenda, editors, *Quo Vadis Quantum Mechanics?*, page 47. Springer, 2005. 1, 3

[CEMM97] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum Algorithms Revisited. *Proc. Roy. Soc. London Ser. A*, 454:339, 1997. 44, 93

[CHSH69] J.F. Clauser, M.A. Horen, A. Shimony, and R.A. Holt. Proposed Experiment to Test Local Hidden-Variable Theories. *Phys. Rev. Lett.*, 23:880, 1969. 14

[CKH98] D. Collins, K.W. Kim, and W.C. Holton. Deutsch-Jozsa Algorithm as a Test of Quantum Computation. *Phys. Rev. A*, 58:R1633, 1998. 44, 59, 61, 67

[CKL01] D.P. Chi, J. Kim, and S. Lee. Initialization-Free Generalized Deutsch-Jozsa Algorithm. *J. Phys. A: Math. Gen.*, 34:5251, 2001. 65, 69

[Cle99] R. Cleve. The Query Complexity of Order-Finding, 1999. LANL e-preprint quant-ph/9911124. 2, 25

[Con62] E.U. Condon. 60 Years of Quantum Physics. *Physics Today*, 15:37, 1962. 6

[CTDL96] C. Cohen-Tannoudji, B. Diu, and F. Laloe. *Quantum Mechanics*. Wiley-Interscience, 1996. 3, 9

[DAK01] K. Dorai, Arvind, and A. Kumar. Implementation of a Deutsch-like Quantum Algorithm Utilizing Entanglement at the Two-qubit Level on an NMR Quantum-information Processor. *Phys. Rev. A*, 63:034101, 2001. 89

[Deh00] H. Dehnen. Quantenmechanik I, 2000. Lecture notes taught at University of Konstanz. Available from: `http://kaluza.physik.uni-konstanz.de/DE/quanten.ps`. 8

[Deu85] D. Deutsch. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proc. R. Soc. Lond. A*, 400:97, 1985. 29, 42, 80

[Dir39] P.A.M. Dirac. A new Notation for Quantum Mechanics. *Proc. Camb. Phil. Soc.*, 35:416, 1939. 3, 11

[DJ92] D. Deutsch and R. Jozsa. Rapid Solution of Problems by Quantum Computation. *Proc. R. Soc. London A*, 439:553, 1992. 25, 34, 38, 44

[Duh91] P. Duhem. *The Aim and Structure of Physical Theory*. Princeton University Press, 1991. 3

[Edm65] J. Edmonds. Paths, Trees and Flowers. *Canadian Journal of Mathematics*, 17:449, 1965. 32

[Ein05] A. Einstein. Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt. *Ann. Phys.*, 17:132, 1905. 3

[Ein07a] A. Einstein. Die Plancksche Theorie der Strahlung und die Theorie der spezifischen Wärme. *Ann. Phys.*, 22:180, 1907. 4

[Ein07b] A. Einstein. Berichtigung zu meiner Arbeit: Die Plancksche Theorie der Strahlung etc. *Ann. Phys.*, 22:800, 1907. 4

[EPR35] A. Einstein, B. Podolski, and N. Rosen. Can Quantum-Mechanical Description of Physical Reality be Considered Complete? *Phys. Rev.*, 47:777, 1935. 10, 13

[ES03] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, Berlin, 2003. 53

[FFL81] R. Freeman, T.A. Frenkel, and M.H. Levitt. Composite Z Pulses. *J. Magn. Reson.*, 44:409, 1981. 86

[FGGS98] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Limit on the Speed of Quantum Computation in Determining Parity. *Phys. Rev. Lett.*, 81:5442, 1998. 65, 81, 93, 94

[FH02] L. Fortnow and S. Homer. A Short History of Computational Complexity, 2002. Available from: http://people.cs.uchicago.edu/~fortnow/papers/history.pdf. 33

[Fis89] G. Fischer. *Lineare Algebra*. Vieweg, 1989. 9

[FP02] W.A. Fedak and J.J. Prentis. Quantum Jumps and Classical Harmonics. *Am. J. Phys.*, 70:332, 2002. 5

[Fuc03] C.A. Fuchs. Quantum Mechanics as Quantum Information, mostly. *J. Mod. Opt.*, 50:987, 2003. 3

[Gis91] N. Gisin. Bell's Inequality Holds for all Non-Product States. *Phys. Lett. A*, 154:201, 1991. 14

[Gro96] L.K. Grover. A Fast Quantum Mechanical Algorithm for Database Search. In ACM, editor, *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, page 212, New York, May 1996. ACM Press. LANL e-preprint quant-ph/9605043. 25

[Gru00] J. Gruska. *Quantum Computing*. Mcgraw Hill Book Co Ltd, 2000. 25, 33, 34

[Hei25] W. Heisenberg. Über die quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen. *Z. Phys.*, 33:59, 1925. 5

[Hei27] W. Heisenberg. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Z. Phys.*, 43:172, 1927. 1

[HGP02] C.H. Holbrow, E. Galvez, and M.E. Parks. Photon Quantum Mechanics and Beam Splitters. *Am. J. Phys.*, 70:260, 2002. 31

[Hir01] M. Hirvensalo. *Quantum Computing*. Natural Computing Series. Springer, Berlin, 2001. 12, 25, 38

[Hod02] A. Hodges. Alan Turing. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2002. Available from: http://plato.stanford.edu/archives/sum2002/entries/turing/. 26

[Hol73] J.H. Holland. Genetic Algorithms and the Optimal Allocation of Trials. *SIAM J. Comput.*, 2:88, 1973. 53

[Hor95] P.J. Hore. *Nuclear Magnetic Resonance*. Oxford University Press, 1995. 18

[HS65] J. Hartmanis and R. Stearns. On the Computational Complexity of Algorithms. *Transactions of the American Mathematical Society*, 117:285, 1965. 31

[IEE94] IEEE, editor. *Proceedings of the 35th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, Silver Spring, MD, USA, Nov. 1994. IEEE Computer Society Press. 115

[Jam66] M. Jammer. *The Conceptual Development of Quantum Mechanics*. McGraw-Hill, 1966. 3, 5

[JK99] J.A. Jones and E. Knill. Efficient Refocusing of One-Spin and Two-Spin Interactions for NMR Quantum Computation. *J. Magn. Reson.*, 141:322, 1999. 20

[Jän01] K. Jänich. *Analysis für Physiker und Ingenieure*. Springer, Berlin, 2001. 7, 9

[KCL98] E. Knill, I. Chuang, and R. Laflamme. Effective Pure States for Bulk Quantum Computation. *Phys. Rev. A*, 57:3348, 1998. 23, 87

[KG00] N. Khaneja and S. Glaser. Cartan Decomposition of $SU(2^n)$, Constructive Controllability of Spin Systems and Universal Quantum Computing. 2000. LANL e-preprint quant-ph/0010100. 58

[Kir83] S. Kirkpatrick. Optimization by Simulated Annealing. *Science*, 220:671, 1983. 52

[Koz92] J.R. Koza. *Genetic Programming: On the Programming of Computers by Natural Selection*. MIT Press, Cambridge, MA, USA, 1992. 53

[KS67] S. Kochen and E. Specker. The Problem of Hidden Variables in Quantum Mechanics. *J. Math. Mech.*, 17:59, 1967. 1

[LB03a] A. Leier and W. Banzhaf. Evolving Hogg's Quantum Algorithm using Linear-Tree GP. In E. Cantú-Paz, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M.A. Potter, A.C. Schultz, N. Jonoska, K.A. Dowsland, J.F. Miller, J.A. Foster, K. Deb, D. Lawrence, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, and G. Kendall, editors, *GECCO-03: Proceedings of the Genetic and Evolutionary Computation Conference, Part I*, volume 2723 of *LNCS*, page 390. Springer, Jul. 2003. 1, 2, 51

[LB03b] A. Leier and W. Banzhaf. Exploring the Search Space of Quantum Programs. In R. Sarker, R. Reynolds, H. Abbass, KC Tan, B. McKay, D. Essam, and T. Gedeon, editors, *Proceedings of the 2003 Congress on Evolutionary Computation*, volume I, page 170, Piscataway, NJ, USA, Dec. 2003. IEEE Computer Society Press. 91

[Lev01] M.H. Levitt. *Spin Dynamics: Basics of Nuclear Magnetic Resonance*. John Wiley & Sons, 2001. 20, 22

[Llo00] S. Lloyd. Quantum Search without Entanglement. *Phys. Rev. A*, 61:010301, 2000. 43

[LP02] W.B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer, Berlin, 2002. 53

[MCS04] P. Massey, J. Clark, and S. Stepney. Evolving Quantum Circuits and Programs through Genetic Programming. In K. Deb, R. Poli, L. Spector, D. Thierens, H.-G. Beyer, A. Tettamanzi, P.L. Lanzi, A. Tyrrell, J. Foster, W. Banzhaf, O. Holland, D. Floreano, E. Burke, M. Harman, P. Darwn, and D. Dasgupta, editors, *Genetic and Evolutionary Computation GECCO-2004*. Springer-Verlag, 2004. 1, 2, 51

[Mer93] N.D. Mermin. Hidden Variables and the two Theorems of John Bell. *Rev. Mod. Phys.*, 65:803, 1993. 1

[Mey00] D.A. Meyer. Sophisticated Quantum Search without Entanglement. *Phys. Rev. Lett.*, 85:2014, 2000. 44

[MFGM01] J.M. Myers, A.F. Fahmy, S.J. Glaser, and R. Marx. Rapid Solution of Problems by Nuclear-Magnetic-Resonance Quantum Computation. *Phys. Rev. A*, 63:032302, 2001. 85

[Mia01] X. Miao. A Polynomial-time Solution to the Parity Problem on an NMR Quantum Computer. 2001. LANL e-preprint quant-ph/0108116. 85

[MRR+53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of State Calculations by Fast Computing Machines. *J. Chem. Phys.*, 21:1087, 1953. 52

[NC00] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 3, 16, 25, 28, 32, 40, 41, 49, 56, 57, 59, 91

[NO02] H. Nishimura and M. Ozawa. Computational Complexity of Uniform Quantum Circuit Families and Quantum Turing Machines. *Theoret. Comput. Sci.*, 276:147, 2002. 41

[Pai79] A. Pais. Einstein and the Quantum Theory. *Rev. Mod. Phys.*, 51:863, 1979. 3

[Pap94] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, 1994. 40

[PBD+00] J. Pan, D. Bouwmeester, M. Daniell, H. Weinfurter, and A. Zeilinger. Experimental Test of Quantum Nonlocality in Three-Photon Greenberger-Horne-Zeilinger Entanglement. *Nature*, 403:515, 2000. 15

[Per95] A. Peres. *Quantum Theory: Concepts and Methods*. Kluwer Academic Publishers, 1995. 3

[Pla00a] M. Planck. Über eine Verbesserung der Wienschen Spektralgleichung. *Verhandl. Dtsch. Phys. Ges.*, 2:202, 1900. 3

[Pla00b] M. Planck. Zur Theorie des Gesetzes der Energieverteilung im Normalspectrum. *Verhandl. Dtsch. Phys. Ges.*, 2:237, 1900. 3

[Rec73] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Fromman-Hozlboog Verlag, Stuttgart, 1973. 53

[RK00] H. Rubens and F. Kurlbaum. *Sitzungsber.Preuss.Akad.Wiss.Phys.-Math.Kl.*, page 929, 1900. 3

[Rub01] B.I.P. Rubinstein. Evolving quantum circuits using genetic programming. In IEEE, editor, *Proceedings of the 2001 Congress on Evolutionary Computation*, page 114, Silver Spring, MD, USA, May 2001. IEEE Computer Society Press. The first version of this paper already appeared in 1999. 1, 2, 51

[Rut11] E. Rutherford. The Scattering of α and β Particles by Matter and the Structure of the Atom. *Phil. Mag.*, 21:669, 1911. 4

[SBBS99] L. Spector, H. Barnum, H.J. Bernstein, and N. Swamy. Quantum Computing Applications of Genetic Programming. In *Advances in Genetic Programming*, volume 3, page 135, 1999. 1, 2, 51, 54, 62, 91

[Sch26a] E. Schrödinger. Quantisierung als Eigenwertproblem (Erste Mitteilung). *Ann. Phys.*, 79:361, 1926. 6

[Sch26b] E. Schrödinger. Quantisierung als Eigenwertproblem (Zweite Mitteilung). *Ann. Phys.*, 79:489, 1926. 6

[Sch26c] E. Schrödinger. Über das Verhältnis der Heisenberg-Born-Jordanschen Quantenmechanik zu der meinen. *Ann. Phys.*, 79:734, 1926. 7

[Sch26d] E. Schrödinger. Quantisierung als Eigenwertproblem (Vierte Mitteilung). *Ann. Phys.*, 81:109, 1926. 7

[Sch35] E. Schrödinger. Die gegenwärtige Situation der Quantenmechanik. *Naturwissenschaften*, 23:807;823;844, 1935. 1

[Sch05] F. Schwabl. *Quantenmechanik für Fortgeschrittene*. Springer, Berlin, 2005. 11

[Sha49] C.E. Shannon. The Synthesis of Two-Terminal Switching Circuits. *Bell Syst. Techn. J.*, 28:59, 1949. 41

[Shi92] Y. Shi. Quantum Lower Bounds for the Collision and the Element Distinctness Problems, 1992. LANL e-preprint quant-ph/0112086. 49

[Sho94] P.W. Shor. Algorithms for Quantum Computation: Discrete Logarithm and Factoring. In IEEE [IEE94], page 124. 1, 2, 25, 46

[Sim94] D.R. Simon. On The Power of Quantum Computation. In IEEE [IEE94], page 116. 25, 44, 46, 47, 60

[SK06] L. Spector and J. Klein. Machine Invention of Quantum Computing Circuits by means of Genetic Programming, 2006. to be published in *Computer-Aided Design*. 51

[Sli90] C.P. Slichter. *Principles of Magnetic Resonance*. Springer, Berlin, 1990. 17, 19

[Sor89] O.W. Sorensen. Polarisation Transfer Experiments in High-Resolution NMR Spectroscopy. *Prog. NMR Spec.*, 21:503, 1989. 18

[Spe04] L. Spector. *Automatic Quantum Computer Programming: A Genetic Programming Approach*. Kluwer Academic Publishers, 2004. 51, 62, 91

[SR95] P. Norvig S.J. Russell. *Artificial Intelligence: Modern Approach*. Prentice Hall, 1995. 52

[SS77] R. Solovay and V. Strassen. A Fast Monte-Carlo Test for Primality. *SIAM J. Comput.*, 6:84, 1977. 28, 32

[SSB] R. Stadelhofer, D. Suter, and W. Banzhaf. Evolving Blackbox Quantum Algorithms using Genetic Programming. to be published in *Computer-Aided Design*. 2

[SSB05] R. Stadelhofer, D. Suter, and W. Banzhaf. Quantum and Classical Parallelism in Parity Algorithms for Ensemble Quantum Computers. *Phys. Rev. A*, 71:032345, 2005. 2

[SW93] M. Schubert and G. Weber. *Quantentheorie: Grundlagen und Anwendungen*. Spektrum Akademischer Verlag, Heidelberg, 1993. 4

[Tur36] A.M. Turing. On Computble Numbers, with an Application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.*, 42:230, 1936. 25, 26

[VSB+01] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, and I.L. Chuang. Experimental Realization of Shor's Quantum Factoring Algorithm using Nuclear Magnetic Resonance. *Nature*, 414:883, 2001. 17

[Vöc02] H. Vöcking. Vorlesung Randomisierte Algorithmen, 2002. Lecture notes taught at University of Dortmund. Available from: http://ls2-www.cs.uni-dortmund.de/lehre/winter200203/randalg/script.ps. 25

[Wae68] B.L. Van Der Waerden. *Sources of Quantum Mechanics*. Dover Publications, 1968. 4

[Weg03] I. Wegener. *Komplexitätstheorie*. Springer Verlag, 2003. 40

[Wei] E.W. Weisstein. Hermitian Operator. In *From MathWorld – A Wolfram Web Resource*. Available from: http://mathworld.wolfram.com/HermitianOperator.html. 7

[WG98] C.P. Williams and A.G. Gray. Automated Design of Quantum Circuits. In C.P. Williams, editor, *Proceedings of the First NASA International Conference on Quantum Computing and Quantum Communications (QCQC)*, volume 1509 of *LNCS*, page 113, New York, Feb. 1998. Springer. 1, 51

[WGC97] W.S. Warren, N. Gershenfeld, and I. Chuang. The Usefulness of NMR Quantum Computing. *Science*, 277:1688, 1997. 23

[WJS+98] G. Weihs, T. Jennewein, C. Simon, H. Weinfurter, and A. Zeilinger. Violation of Bell's Inequality under Strict Einstein Locality Conditions. *Phys. Rev. Lett.*, 81:5039, 1998. 1, 15

[Zac03] R. Zach. Hilbert's Program. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2003. Available from: http://plato.stanford.edu/archives/fall2003/entries/hilbert-program/. 26

[ZB02] M. Zukowski and C. Brukner. Bell's Theorem for General N-Qubit States. *Phys. Rev. Lett.*, 88:210401, 2002. 14

[ZBLW02] M. Zukowski, C. Brukner, W. Laskowski, and M. Wiesniak. Do All Pure Entangled States Violate Bell's Inequalities for Correlation Functions? *Phys. Rev. Lett.*, 88:210402, 2002. 15

[Zei99a] A. Zeilinger. A Foundational Principle for Quantum Physics. *Foundations of Physics*, 29:631, 1999. 1, 3

[Zei99b] A. Zeilinger. Experiment and the Foundations of Quantum Physics. *Rev. Mod. Phys.*, 71:288, 1999. 14

[ZWJA05] A. Zeilinger, G. Weihs, T. Jennewein, and M. Aspelmeyer. Happy Centenary, Photon. *Nature*, 433:230, 2005. 1, 14

Index