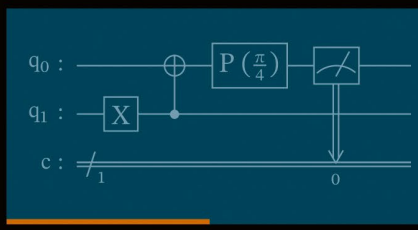
Kaveh Bashiri

Mit Codebeispielen zur direkten Umsetzung




```
from qiskit import QuantumCircuit
from qiskit.quantum_info.operators import Operator
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
import numpy as np

omega='1101'# Auswahl der Nadel im Heuhaufen

# Schritt (1)
n = len(omega)
omega= int(omega,2)
# Ab jetzt behandeln wir
# omega wie eine ganze Zahl
circ = QuantumCircuit(n,n)
circ.h(range(n))

# Schritt (2)
# Implementierung der Matrix zugehörig zu U_(f_\omega)
oracle_omega = np.identity(2**n)
oracle_omega[omega,omega]=-1

# Implementierung der Matrix zugehörig zu U_(f_0)
oracle_null = -np.identity(2**n)
oracle_null[0,0]=1

# Grover-Iterationen
r = int(np.floor(np.pi/4*np.sqrt(2**n)))
for _ in range(r):
```

QUANTEN COMPUTING

GRUNDLAGEN · ALGORITHMEN · ANWENDUNG

Von den Grundlagen bis zum Code: Quantentheorie verstehen

Superposition, Verschränkung, Qubits und das Quantenschaltkreismodell

Quantenalgorithmen mit Qiskit implementieren

Rheinwerk Computing

Inhalt

Kapitel 1
Einleitung

1.1 Die Grenzen des klassischen Computers

Das Leben verändert sich immer, aber es scheint, dass die letzten Jahrzehnte besonders viele und tiefgreifende Änderungen mit sich gebracht haben, denn Computer sind aus so gut wie keinem Bereich mehr wegzudenken. Unser Berufsalltag, unsere Freizeit, quasi unser ganzes Leben wird durch Algorithmen beeinflusst, die uns begleiten, wenn wir online Versicherungen vergleichen, Urlaube buchen oder Freunden auf Social Media folgen.

Das ist nur möglich, weil auf der ganzen Welt gigantische Rechenzentren und IT-Infrastrukturen entstanden sind, die unglaubliche Datenmengen verarbeiten. Viel Geld und viel Aufwand werden investiert, um sie immer effizienter zu machen; die Technik steht nie still. Diese Optimierungen schaffen es sogar bis in Ihre Hosentasche: Unsere Smartphones können mit jeder Generation mehr. Rechenleistung, die man vor einigen Jahren noch auf Hochleistungsrechnern suchen musste, passt nun in unsere Hand. Ob es um künstliche Intelligenz geht, Cloud-Computing oder IoT: Alle bahnbrechenden, disruptiven Ideen der letzten Jahre beruhen darauf, dass immer mehr Rechenleistung und Speicherplatz zur Verfügung stehen.

Doch an gewissen Punkten stoßen die gegenwärtigen Methoden an ihre Grenzen. Beispielsweise gibt es physikalische Systeme, die so komplex sind, dass sie nicht realistisch simuliert werden können. Die Wissenschaft steht vor immer mehr Problemen, die sich nicht einfach durch mehr Rechenzeit lösen lassen.

Zudem ist es sinnvoll anzunehmen, dass wir am Ende der Gültigkeit des *Moore'schen Gesetzes* angekommen sind und dass diese Prognose in Zukunft nicht mehr gilt. Dieses Gesetz besagt nämlich, dass sich die Dichte der Bauelemente in einem aktuellen, dem Stand der Technik entsprechenden Computerchip regelmäßig verdoppelt und somit exponentiell steigt. Diese Dichte ist eine Kenngröße für die mögliche Rechenleistung aktueller Computer. Je mehr Transistoren auf einem Chip untergebracht werden können, desto größer ist seine Rechenleistung.

Doch warum scheint dieses Gesetz jetzt nicht mehr realistisch zu sein, während wir in den letzten Jahrzehnten doch tatsächlich ein derartiges exponentielles Wachstum der Rechenleistung beobachten konnten?

Die Antwort auf diese Frage zwingt uns zum ersten – und bei weitem nicht zum letzten (!) – Mal in diesem Buch, an die Grenzen unseres menschlichen Verständnisses der Realität zu gehen. Es ist nämlich so, dass eine höhere Dichte an Bauelementen impliziert, dass diese Bauelemente immer kleiner werden müssen. Auf der Ebene der kleinsten Elemente – d. h. auf der mikroskopischen Skala – sieht die Wirklichkeit ganz anders aus, als wir sie wahrnehmen. Hier herrschen die Gesetze der *Quantenmechanik*! Sie ist faszinierend und erschreckend zugleich, denn wir können auf dieser Ebene Phänomene entdecken, die wir aus unserer realen Umgebung in der makroskopischen Welt nicht kennen. Und da unsere gegenwärtigen Computer makroskopische Objekte mit makroskopischen Bauelementen sind, werden sie auf der mikroskopischen Skala in die Knie gezwungen. Daher bezeichne ich auch fortan diese Computer als *klassische Computer* und die zugehörige Wissenschaft als *klassische Informatik*, um zu verdeutlichen und abzugrenzen, dass diese nur für die bisher übliche, makroskopische und nicht auf der Quantenmechanik beruhenden Welt anwendbar ist.

War es das jetzt? Sind das die Grenzen der Berechenbarkeit, die wir als Menschen erreichen können? Heißt es, dass wir einige Probleme (wie die Simulation gewisser physikalischer Systeme) niemals lösen können, und dass ab irgendeinem Zeitpunkt das Moore'sche Gesetz nicht mehr gilt?

Nein, für unsere Reise ist das erst der Anfang! Denn – beginnend mit Arbeiten aus den 1980er-Jahren von Paul Benioff, Yuri Manin und Richard Feynman (siehe [Ben80], [Man80], [Fey82] oder [NC10, Abschnitt 1.1.1]) – wurde eine Alternative zur klassischen Informatik entwickelt, die nicht nur keine Angst vor der Quantenmechanik hat, sondern vielmehr ihre gesamte Stärke aus ihr bezieht: die *Quanteninformatik*! Sie ist ein völlig neuer Ansatz, Berechnungen durchzuführen, in denen quantenmechanische Phänomene bewusst ausgenutzt werden, um für gewisse, bedeutsame Probleme signifikante Effizienzvorteile zu erreichen, wenn man sie mit der klassischen Informatik vergleicht. Den Rechner, der diese auf quantenmechanischen Phänomenen basierende Berechnungen durchführt, nennt man *Quantencomputer*.

Diese neue Form der Informatik wird (wahrscheinlich) unsere digitale Zukunft maßgeblich verändern. Es lohnt sich daher, sich mit ihr zu beschäftigen, und zwar schon heute. Jetzt ist der richtige Zeitpunkt für einen Einstieg in dieses Thema, damit Sie für diese Umstellung gewappnet sind und den Weg

mitgestalten können, auch wenn Quantencomputer bisher die »alltägliche« Informatik noch kaum berühren.

Wenn Sie bereit sind, dann ist dieses Buch der erste Schritt für Sie. Sein Ziel ist es, Ihnen einen praxisorientierten Einstieg in die Quanteninformatik zu geben. Im folgenden Abschnitt möchte ich Ihnen als Motivation schon mal einige Grundideen der Quanteninformatik aufzeigen. Wie in einem kurzen Trailer soll er zeigen, was Sie in diesem Buch erwarten können. Die faszinierende und mitunter anspruchsvolle Reise beginnt erst später.

1.2 Vom klassischen Computer zum Quantencomputer – Grundideen

In der klassischen Informatik werden *Bits* in sogenannten *Bit-Registern* gespeichert und verarbeitet. Diese Bits können die Werte 0 oder 1 annehmen und sind die kleinsten Informationseinheiten der klassischen Informatik. In diesen Bit-Registern werden mittels *Algorithmen*, die nichts anderes sind als eine Abfolge eindeutig definierter Rechenschritte, Berechnungen durchgeführt und Probleme gelöst.

Auch in der Quanteninformatik möchte man in Registern Informationseinheiten verarbeiten. Wie in der klassischen Informatik will man Berechnungen durchführen und Probleme lösen. Der fundamentale Unterschied in der Quanteninformatik ist nun, dass man als elementare, kleinste Basiseinheit keine Bits verwendet, sondern sogenannte *Qubits*.

1.2.1 Was ist ein Qubit?

Ein Qubit ist zunächst einmal nichts anderes als ein sehr spezielles physikalisches System, in dem quantenmechanische Gesetze gelten. Als solches hat ein Qubit gewisse physikalische Eigenarten. Eine der größten Errungenschaften des 20. Jahrhunderts war es, diesen Eigenarten auf die Spur zu kommen. Später in diesem Kapitel möchte ich auf diese Historie kurz eingehen.

Diese physikalischen Eigenschaften des Qubits werden mathematisch durch eine *Wellenfunktion* oder – wie wir sagen werden – einen *Zustandsvektor* (oder kurz: *Zustand*) beschrieben. Man modelliert Qubits nun so, dass zwei der möglichen Zustandsvektoren eine besondere Bedeutung haben. Diese beiden Zustände notiert man in der sogenannten *Dirac-Notation* als $|0\rangle$ und $|1\rangle$ (gesprochen »Ket-Null« bzw. »Ket-Eins«), und sie übernehmen die Rolle der

Bitwerte 0 und 1 aus der klassischen Informatik. Genauer gesagt meinen wir damit, dass wir

- einem Qubit die Information 0 zuordnen, wenn der Zustand des Qubits gegeben ist durch $|0\rangle$, und
- einem Qubit die Information 1 zuordnen, wenn der Zustand des Qubits gegeben ist durch $|1\rangle$.

Somit wissen Sie nun, wie Sie die Zustände eines Qubits deuten können. Sie werden in diesem Buch auch lernen, dass wir diese Qubits auch bewusst steuern können (nämlich via *unitärer Matrizen*) und auslesen können (via *Messungen*). Noch ist aber nicht ersichtlich, worin der versprochene Vorteil in der Quanteninformatik – auch *Quantenüberlegenheit* genannt – bei gewissen Problemen besteht. Dieser Vorteil entsteht durch die Ausnutzung zweier besonderer Eigenarten aus der Quantenmechanik: das *Superpositionsprinzip* und die *Verschränkung*. Es stellt sich heraus, dass diese beiden Eigenschaften essenziell sind für die versprochene Quantenüberlegenheit. Daher möchte ich kurz auf diese beiden Eigenschaften eingehen und in einfachen Worten beschreiben, wieso diese zum Vorteil führen können.

1.2.2 Das Superpositionsprinzip

In der klassischen Informatik werden in den Bit-Registern nur die beiden Werte 0 oder 1 verarbeitet. Es ist ein fundamentaler Unterschied zwischen der Quanteninformatik und der klassischen Informatik, dass Qubits auch Zustände annehmen können, die weder gleich $|0\rangle$ noch gleich $|1\rangle$ sind, aber von beiden Zuständen beeinflusst sind. Damit meint man, dass Qubits Zustände annehmen können, bei denen nach dem Auslesen – d.h. nach der Messung – nur mit einer gewissen *Wahrscheinlichkeit* gesagt werden kann, ob der Zustand $|0\rangle$ und $|1\rangle$ angenommen wird. Man kann bei solchen Zuständen nicht mit Sicherheit sagen, in welchem der beiden Zustände sie sind und welchen der beiden Zustände sie nach der Messung annehmen werden.

Insbesondere heißt das, dass Qubits in solchen Zuständen sich vor der Messung nicht entscheiden müssen, ob sie den einen oder den anderen Zustand annehmen. Man sagt, dass die Qubits beide Zustände *überlagert* annehmen. Solche Zustände bezeichnet man als *Superposition* der beiden Zustände $|0\rangle$ und $|1\rangle$. Dies widerspricht zunächst unserer Intuition. Denn wie kann ich beispielsweise in einer Überlagerung – also in einer Art Mischung – von zwei Orten sein? Für uns ist das undenkbar, aber für Quantenobjekte ist das kein Problem!

Mit einem Qubit, das sich in Superposition befindet, ließe sich also so rechnen, als würde man parallel sowohl vom Zustand $|0\rangle$ als auch vom Zustand $|1\rangle$ aus die Berechnung beginnen. Man muss also nicht einzeln für beide Startpunkte jeweils die Berechnungen durchführen, sondern kann beide Startpunkte *parallel* berücksichtigen. Diese Beobachtung klingt nach einem besonders hilfreichen Merkmal für die Quanteninformatik, da sich so möglicherweise viele Berechnungen parallelisieren ließen und somit effizienter durchgeführt werden könnten.

Stellen Sie sich vor, dass Sie vor einem Kleiderschrank stehen, ein passendes Hemd suchen und zwischen zwei Optionen schwanken.

Mit einem klassischen Computer müssten Sie jedes Hemd einzeln anprobieren und anschließend das bessere Hemd auswählen. Mit einem Quantencomputer könnten Sie (ausgehend von einer Superposition dieser Hemden) beide Hemden parallel anprobieren und anschließend die Entscheidung treffen. Das ist ein Phänomen, das man in der Quanteninformatik auszunutzen vermag und das bereits andeutet, wieso Quantencomputer beim Lösen gewisser Probleme Effizienzvorteile haben.

1.2.3 Verschränkung

Selbst Albert Einstein, ein Mensch, der mit seinem Intellekt sowie seinen bahnbrechenden wissenschaftlichen Erfolgen Geschichte geschrieben hat, gab diesem Phänomen das Attribut »spukhaft«.

Daher bitte ich Sie, nachsichtig mit sich selbst zu sein, falls Sie bei der folgenden Beschreibung dieses Phänomens noch skeptischer oder verwirrter sind als beim Superpositionsprinzip.

Im Jahre 1935 wurde in einer wissenschaftlichen Arbeit (unter Beteiligung von Albert Einstein) als direkte Konsequenz der bis dahin weit etablierten Gesetze der Quantenmechanik ein Phänomen beschrieben, das Jahre später von Erwin Schrödinger den Namen *Verschränkung* bekam. Es besagt, dass miteinander verschränkte Quantenobjekte eine Wirkung aufeinander ausüben können, ohne dass dabei die räumliche Distanz zwischen diesen eine Rolle spielt. In der Theorie wäre es somit denkbar, dass wenn zwei miteinander verschränkte Quantenobjekte sich beliebig weit voneinander entfernen – sagen wir, dass diese sich in verschiedenen Galaxien befinden –, gewisse Manipulationen des einen Quantenobjekts direkte Auswirkungen auf das andere Quantenobjekt haben. Dies ist natürlich für uns unvorstellbar. Jedenfalls stellen wir uns eine Wirkung auf ein anderes Objekt stets so vor, dass diese entweder auf direktem Kontakt beruht oder durch ein Medium weitergelei-

tet wird. Dieser Kontakt – egal ob direkt oder indirekt – kann höchstens so schnell wie Licht sein. Diese Eigenschaft nennt man *Lokalismus*. Aber – schon wieder – ist die Realität auf der Quantenebene anders!

Und wie kann man diese Eigenschaft in der Quanteninformatik ausnutzen? Ganz einfach: indem man Qubits miteinander verschränkt und dann den Umstand ausnutzt, dass Manipulationen des einen Qubits unmittelbare Auswirkungen auf das andere Qubit haben. Es gilt sogar noch mehr: Die Ausnutzung der Verschränkung ist notwendig für *signifikante Quantenüberlegenheit*, wobei ich hier die Quantenüberlegenheit als signifikant bezeichne, wenn die Laufzeit der Lösung eines Problems auf einem Quantencomputer exponentiell schneller ist als die entsprechende Laufzeit auf einem klassischen Computer. Für mehr Details zu diesem Resultat verweise ich auf [JL03].

1.2.4 Quantenüberlegenheit

Nun haben Sie gesehen, dass sich die klassische Informatik und die Quanteninformatik fundamental unterscheiden. Ich habe auch erwähnt, dass – zumindest auf dem Papier – ein Quantencomputer aufgrund des Superpositionsprinzips und der Verschränkung einen klassischen Computer schlagen sollte (zumindest in einigen Disziplinen).

Doch ist dem wirklich so?

In der Tat! In diesem Buch werden Sie Quantenalgorithmen (beispielsweise den Shor-Algorithmus) kennenlernen, die einzelne Probleme exponentiell schneller als die besten bekannten Algorithmen auf einem klassischen Computer lösen können. Damit Sie diese Algorithmen sowohl in der Theorie verstehen als auch praktisch anzuwenden lernen, habe ich dieses Buch geschrieben.

1.3 Auswirkungen, Nutzen und Gefahren eines Quantencomputers für die Gesellschaft

Wenn es um bahnbrechende Innovationen geht, sollte man sich stets Gedanken machen, welche Auswirkungen sie auf unser Leben und unsere Gesellschaft haben. Es steht außer Frage, dass ein leistungsfähiger Quantencomputer unser Leben verändern wird. Auf einige dieser Aspekte möchte ich im Folgenden eingehen.

Auf der positiven Seite

»Für den Quantencomputer wird mittel- und langfristig ein Marktpotenzial von bis zu dreistelligen Milliardenbeträgen prognostiziert« heißt es in

[Kag20]. Dieses Potenzial beruht auf der Optimierung von Lieferketten und der Lösung von Logistikproblemen. Außerdem werden Quantenrechner bei der Herstellung von Medikamenten und Chemikalien helfen. Des Weiteren ist die Ausnutzbarkeit des *Quantum Machine Learnings*, d. h. der Kombination von Quanteninformatik und künstlicher Intelligenz denkbar (siehe [Bun22]). Mit der Entwicklung von leistungsstarken Quantencomputern sind also positive Effekte verbunden, die auch wirtschaftlich ausgenutzt werden können. Daher liegt die Forschung und Entwicklung in diesem Bereich ebenso im Interesse vieler Firmen und Forschungsinstitute.

Auswirkungen auf die Kryptografie

Die direktesten und konkretesten Auswirkungen leistungsstarker Quantencomputer zielen auf die IT-Sicherheit ab, genauer gesagt auf die Kryptografie. Hier liegen daher auch die größten Gefahren.

Ein wesentlicher Teil der Kryptografie basiert nämlich auf sogenannten *Public-Key-Verfahren*. Die Sicherheit heute gängiger Public-Key-Verfahren beruht aber darauf, dass gewisse mathematische Probleme schwierig zu lösen sind – zumindest vermutlich …

Die zwei wichtigsten mathematischen Probleme, auf denen diese Verfahren aufbauen, sind das sogenannte *Faktorisierungsproblem* und das *diskrete Logarithmusproblem*. Es hat sich in den letzten Jahrzehnten herausgestellt, dass diese Probleme schwierig zu lösen sind. Genauer gesagt heißt das, dass noch niemand einen effizienten Algorithmus für diese beiden Probleme gefunden hat.

Jedenfalls nicht auf einem klassischen Computer! Diese Einschränkung ist sehr wichtig, denn man hat in den 1990er-Jahren herausgefunden, dass ein leistungsstarker Quantencomputer diese beiden Probleme sehr wohl und sehr effizient lösen kann (Stichwort: *Shor-Algorithmus*). Somit sind aktuelle IT-Systeme, deren Sicherheit zum überwältigenden Teil auf diesen beiden mathematischen Problemen basiert, nur deswegen noch sicher, weil niemand (insbesondere nicht die Nachrichtendienste) einen leistungsstarken Quantencomputer im Keller hat.

Viele Jahre lang lebte es sich auch ganz gut mit diesem Risiko. Doch immense Investitionen und wissenschaftliche Errungenschaften befeuern die Sorge, dass tatsächlich so ein leistungsstarker Quantencomputer existieren wird. Daher entstand in den letzten Jahren ein hochspannendes Forschungsgebiet: die *Post-Quanten-Kryptografie*, in der kryptografische Verfahren entwickelt werden, von denen man vermutet, dass sie selbst einem leistungsstarken Quantencomputer standhalten können.

In diesem Buch werde ich mich auf die *Quantenalgorithmen* fokussieren – d.h. auf die Algorithmen, die auf einem Quantencomputer laufen. Auf die Post-Quanten-Kryptografie werde ich jedoch am Ende dieses Buchs in Kapitel 13 kurz eingehen.

1.4 Quanteninformatik und die Realität

Ein Problem, das wir bisher erfolgreich umschifft haben, ist die Frage, wie man in der Praxis so einen Quantencomputer baut. Eine direkte Folgefrage, die wir uns auch sofort stellen müssten, ist, wann wir mit so einer Quantencomputer-Revolution rechnen müssen.

Dies sind zwei essenzielle Fragen, die in der Wissenschaft aktuell intensiv untersucht werden und die wir mit Sicherheit als die *Flaschenhälse der Quanteninformatik* identifizieren können. Auf sie werde ich jedoch nicht eingehen – wenn Sie programmieren lernen wollen, müssen Sie sich ja auch nicht erst damit beschäftigen, wie ein Transistor hergestellt wird oder wie eine Festplatte funktioniert. Der Fokus unserer Überlegungen wird daher auf der Frage liegen, wie sich die Ideen der Quantenmechanik auf die Informatik auswirken und wie sie sich in Programmcode umsetzen lassen. Aufgrund der Bedeutung des praktischen Aspekts möchte ich aber sehr kurz die vielversprechendsten Ansätze, die größten Herausforderungen sowie die wahrscheinlichsten Prognosen zum bzw. beim Bau eines Quantencomputers ansprechen. Hierdurch lernen Sie zumindest die wichtigsten Begriffe kennen und können anhand der Referenzen tiefer in die jeweiligen Themen einsteigen.

Welche Ansätze gibt es?

Es gibt eine Vielzahl verschiedener Ansätze, einen Quantencomputer in der Praxis zu realisieren. Wie bereits erwähnt, haben solche Rechner große Auswirkungen auf die IT-Sicherheit. Genau deshalb hat das *Bundesamt für Sicherheit in der Informationstechnik (BSI)* eine fortlaufende Studie in Auftrag gegeben, in der diese Entwicklungen beobachtet und bewertet werden (siehe [Bun23a]). In diese Studie fließen aber verständlicherweise nur Erkenntnisse ein, die öffentlich publiziert wurden und ohne Einschränkungen zugänglich sind. Die Bewertung der Ansätze erfolgt anhand eines in dieser Studie festgelegten Bewertungsschemas für die potenziellen Hardware-Plattformen für einen Quantencomputer.

Gemäß der aktuellen Fassung dieser Studie sind die folgenden drei Ansätze die vielversprechendsten und am weitesten fortgeschrittenen Plattformen für einen Quantencomputer:

- **Ionenfallen:** Die Idee bei Ionenfallen-basierten Plattformen ist es, geladene Teilchen – d.h. *Ionen* – in einem elektrischen Feld einzufangen und dann via Laser, Magnetfelder oder Mikrowellenfelder zu manipulieren. Die möglichen Energieniveaus der Ionen bilden hierbei die besonderen Zustände des Qubits (d.h. die Zustände $|0\rangle$ und $|1\rangle$).
- **Supraleiter:** Bei Quantencomputern, die auf Supraleitern basieren, werden supraleitende Schaltkreise genutzt, um Qubits zu realisieren. Dabei bildet die Flussrichtung des Stroms die besonderen Zustände des Qubits.
- **Neutrale Atome:** Bei dieser Plattform werden die äußeren Elektronen eines Atoms mittels Laser angeregt und auf dem angeregten Zustand mittels sogenannter *optischer Pinzetten* gefangen. Somit bleiben die Atome ungeladen. Die verschiedenen Energieniveaus der äußeren Elektronen definieren die besonderen Zustände des Qubits.

Für weitere Details zu den Ideen hinter diesen Ansätzen siehe [Bun23a]. Diese einzelne Plattformen haben jeweils Vorteile und Nachteile. Beispielsweise sind Supraleiter-Plattformen schnell und leichter skalierbar, während Ionenfallen-Plattformen länger stabil gehalten werden können (bezüglich der sogenannten *Dekohärenz*, siehe Abschnitt 4.4.4) und Neutrale-Atome-Plattformen effizient Interaktionen zwischen den Qubits umsetzen können.

Was sind die Herausforderungen?

Quantensysteme sind sehr sensibel. Sobald sie mit ihrer Umwelt in Kontakt kommen, verändern sie sich. Und das ist natürlich etwas, was man bei einem Quantencomputer nicht gebrauchen kann, da man hier mittels solcher Quantensysteme präzise Berechnungen durchführen möchte. Denn solche unkontrollierten Veränderungen haben natürlich eine direkte Konsequenz für den Zustand der Quantensysteme und führen folglich zu Fehlern bei den Berechnungen.

Wir haben es also mit hochsensiblen Objekten zu tun. Kein Problem, dann müssen wir diese eben isolieren und alle Störeinflüssen vermeiden.

Es stellt sich aber heraus, dass Quantensysteme stets Einflüssen und Störungen aus der Umgebung (zum Beispiel durch kosmische Strahlung) ausgesetzt sind. Die aus diesen äußeren Störeinflüssen resultierenden Fehler müssen somit in der Praxis immer mit berücksichtigt werden.

Im hochspannenden und aktuell sehr lebendigen Forschungsfeld der *Quantenfehlerkorrektur* versucht man – zum Preis eines gewissen *Overheads* (d.h. eines Zuwaches an Größe und Rechenzeit) –, diese unausweichlichen Fehler aktiv parallel zu den Berechnungen zu korrigieren (siehe Abschnitt 8.5 für mehr Details).

Zu erwähnen ist außerdem noch der sogenannte *Noisy-Intermediate-Scale-Quantum-Ansatz* oder abgekürzt *NISQ-Ansatz*, bei dem versucht wird, die Störeinflüsse zu akzeptieren und stattdessen sich auf spezielle Probleme und Quantenalgorithmen zu fokussieren, die trotz der unausweichlichen Fehler Ergebnisse liefern können (siehe [Pre18] oder [Bun23a] für mehr Details).

Wann ist es so weit?

Die letzten Gedanken klingen nicht hoffnungsvoll und lassen nicht erwarten, dass wir in nächster Zeit bahnbrechende Quantenalgorithmen auf leistungsstarken Quantencomputern laufen lassen können, denn die Hindernisse in der Umsetzung scheinen aktuell noch zu hoch zu sein. Doch hohe Investitionen in die Forschung und einzelne Entwicklungssprünge und Meilensteine lassen vermuten, dass es zwar heute noch zu früh ist, dass es aber in den nächsten Jahren – falls sich die Entwicklung derart fortsetzt – mit Quantencomputern durchaus ernst werden kann.

Die Frage nach dem *Q-Day* – d.h., dem Tag, an dem ein nach gewissen Kriterien leistungsstarker Quantencomputer existiert – ist Gegenstand wichtiger Untersuchungen. Beispielsweise wird in der bereits erwähnten Studie [Bun23a] oder in [MP24] eine Prognose (basierend auf dem aktuellen Stand der Technik) abgegeben. Grob zusammengefasst deuten diese Prognosen an, dass eine signifikante (und somit nicht vernachlässigbare) Wahrscheinlichkeit existiert, dass in den nächsten Jahrzehnten ein derartiger Quantencomputer zur Verfügung steht. Gegeben der möglichen gravierenden Auswirkungen eines solchen Quantencomputers ist so eine Aussage genug, um sich – nicht zuletzt aufgrund eines rigorosen Risikomanagements – intensiv mit der Quanteninformatik zu beschäftigen. Und das nicht nur, weil es sich um ein höchst elegantes und spannendes Forschungsfeld handelt, bei dem viele wesentlichen und spannenden Fragestellungen noch unbeantwortet sind.

Da Sie sich diesem Buch widmen, haben Sie bereits den ersten Schritt getätigt, um für diese Zukunft gewappnet zu sein und diese möglicherweise mitgestalten zu können. Ich freue mich, Sie auf dieser Reise begleiten zu dürfen.

1.5 Über dieses Buch

Sie haben nun eine erste Vorstellung davon, warum man sich mit der Quanteninformatik beschäftigen sollte. In diesem Kapitel möchte ich Ihnen die Rolle dieses Buches erläutern und zeigen, wie es Ihnen hilft, die ersten Schritte in diesem faszinierenden Thema zu gehen.

Ich fange mit den Zielen des Buchs an. Anschließend möchte ich Ihnen die pädagogische Herangehensweise erläutern, damit Sie wissen, in welcher Ge-

stalt dieses Buch versucht, Ihnen das mitunter abstrakte Thema näherzubringen. Danach werde ich die inhaltlichen Voraussetzungen thematisieren, die für dieses Buch notwendig sind.

1.5.1 Ziele

Das große *übergeordnete Ziel* dieses Buchs ist ein niederschwelliger, kleinschrittiger und praxisorientierter Einstieg in die Quanteninformatik. Hierzu muss als notwendige Grundlage zunächst ein theoretisches Verständnis für die Quantenmechanik erarbeitet werden. Anschließend soll dieses Verständnis genutzt werden, damit Sie Quantenalgorithmen nachvollziehen, selbst formulieren und via Qiskit implementieren können. Im Detail sollen Ihnen durch dieses Buch folgende Inhalte und Fähigkeiten vermittelt werden:

- Sie erlangen ein *grundlegendes Verständnis für die Theorie der Quantenmechanik*.
- Bezüglich der *physikalischen Aspekte der Quantenmechanik verfügen Sie über elementare Kenntnisse*, die Sie mit weiterführender Literatur weiter vertiefen können.
- Sie entwickeln eine *Intuition für Quantenalgorithmen* und können diese verstehen und (in einfachen Szenarien) selbst formulieren.
- Sie wissen, wie man *Quantenschaltkreise konstruiert*.
- Sie verfügen über die Fähigkeiten, Quantenalgorithmen in einfachen Beispielen *konkret via Qiskit implementieren* zu können.
- Sie gewinnen einen guten *Überblick über einige bahnbrechende Quantenalgorithmen* (wie etwa Shor und Grover).
- Alle erlernten Inhalte können Sie in weiterführender Literatur vertiefen und erweitern.

Den letzten Punkt möchte ich gerne noch weiter ausführen: Ich beabsichtige, die theoretische Detailtiefe und den Abstraktionsgrad derart zu gestalten, dass Ihr Einstieg in weiterführende Literatur möglichst stetig und reibungslos erfolgt, wenn Sie sich tiefer mit der Materie beschäftigen. Anders ausgedrückt: Ich möchte bei den theoretischen Konzepten den *Formalismus und die Abstraktion so gering wie möglich – aber nicht noch geringer – halten*.

Genauer gesagt bedeutet dies, dass ich einzelne Konzepte so formulieren werde, dass diese auf andere Situationen verallgemeinert werden können, sodass *keine vollständige Wiederaufarbeitung des Konzepts* notwendig ist. Letzteres ist oftmals nötig, wenn man in einer Präsentation die Dinge so sehr auf die konkrete Situation vereinfacht hat, sodass das übergeordnete Konzept nicht

mehr sichtbar ist oder man dieses übergeordnete Konzept nicht mehr durch Extrapolation der konkreten Situation erhalten kann. Diese Strategie hat den Preis, dass ich an einzelnen Stellen Konzepte in abstrakterer Form einführen werde, als es für die konkreten Beispiele dieses Buchs notwendig ist.

1.5.2 Herangehensweise

In dieses Buch sind meine persönlichen Erfahrungen aus der universitären Lehre eingeflossen. Sie zeigen, dass die folgende Herangehensweise hilfreich für einen langfristigen Lernerfolg ist:

- Es ist wichtig, dass die erarbeiteten Konzepte nicht vom Himmel fallen. Daher beabsichtige ich, *alle Definitionen (insbesondere die Postulate der Quantenmechanik) und Resultate intuitiv herzuleiten*. Diese Herleitung basiert auf einem mathematischen, physikalischen Grundverständnis.
- Wenn man sich in sehr umfassende Themen einarbeitet, ist es unvermeidlich, dass es einzelne Aspekte gibt, die anfangs noch nicht ersichtlich sind und vielleicht sogar verwirren. Typischerweise werden diese Aspekte erst im weiteren Verlauf der Reise ersichtlich. Beispielsweise werden Sie sich zu Beginn unserer Einarbeitung in die Quantenmechanik fragen, wieso man überhaupt mit komplexen Zahlen hantieren muss.

  Mitunter gibt es Studierende, die sich an solchen Stellen abgehängt fühlen und denken: »Eigentlich müsste ich das jetzt bestimmt verstehen!« Dies kann zu Frustration führen, was natürlich den Lernerfolg beeinträchtigt. Diese Situationen möchte ich vermeiden, indem ich *potenzielle Unklarheiten, die erst im weiteren Verlauf aufgeklärt werden, benenne und auf zukünftige Kapitel verweise*.

  Für manchen kann es vielleicht so sein, dass zu viele Fragen aufgeworfen werden, die erst in späteren Kapiteln beantwortet werden, Auch dies betrachte ich als berechtigten Einwand. Daher beabsichtige ich, die aufgeworfenen Fragen mit konkreten Verweisen zu den jeweiligen Antworten zu versehen. Daher bitte ich Sie, direkt zur Antwort zu blättern, wenn Sie die Antworten lieber sofort haben möchten.
- Ich versuche so oft wie möglich, *Beispiele und Metaphern einzuführen*, die man sich vor dem geistigen Auge vorstellen kann. So können Sie die Konstruktionen besser in Erinnerung behalten. Stumpfes Auswendiglernen führt meiner Erfahrung nach zu keinem langfristigen Lernerfolg.
- Allgemein werde ich versuchen, an so vielen Stellen wie möglich *konkrete Rechen- und Codebeispiele* einzubauen. Bei den Beispielen lege ich Ihnen nahe, erst selbst zu versuchen, sie durchzurechnen oder zu implementie-

ren. Ich bin überzeugt, dass ein langfristiger Lernerfolg nur durch die aktive Arbeit an einem Thema möglich ist – und die beginnt mit einem leeren Blatt Papier. Eine Sprache erlernen Sie ja auch nicht nur durch Zuhören. Sie müssen diese neue Sprache auch sprechen – auch wenn es anfangs nur mühsam funktioniert.

- Mein Ziel ist also, die Inhalte mit Ihnen gemeinsam zu erarbeiten. Sie finden daher am Ende der Kapitel Übungsaufgaben, mit denen Sie wichtige Punkte wiederholen und vertiefen können. Die Lösungen zu diesen Aufgaben finden Sie online bei den Materialien zum Buch unter: *https://www.rheinwerk-verlag.de/5808*
- Dort finden Sie auch Jupyter Notebooks, die den Code der Beispiele enthalten. Wenn Sie Qiskit installiert haben (siehe Abschnitt 1.6), können Sie diese Beispiele direkt ausführen und müssen nicht den Code aus dem Buch abtippen.

1.5.3 Voraussetzungen

Wenn man stets bei null anfängt, ist es schwierig, Fortschritte zu machen. Dies liegt daran, dass dann die Vorbereitungsarbeit schon so viel Raum einnehmen muss, sodass für das Eigentliche, das Neue, weniger Raum zur Verfügung steht. Daher wird es inhaltliche Voraussetzungen geben, damit wir das Thema angemessen behandeln können. Das ändert aber natürlich nichts daran, dass es mir wichtig ist, dass so viele Menschen wie möglich so einfach wie möglich einen Einstieg in dieses Thema finden.

Vorwissen aus der Mathematik

Quantenmechanik hat den Ruf, kompliziert und eigentlich mit dem gesunden Menschenverstand nicht vereinbar zu sein. Letzteres ist eine philosophische Debatte, die ich hier ignoriere. Aber ein persönliches Ziel in diesem Buch ist es, Ihnen zu zeigen, dass Ersteres (zumindest für unsere Zwecke) nicht wahr ist. Das wichtigste Werkzeug ist natürlich die Mathematik. Wir werden viel Mathematik brauchen, das ist unvermeidbar. Aber da wir die Quantenmechanik nur für die Quanteninformatik benötigen, kann ich an vielen Stellen die Theorie vereinfachen. Trotzdem ist die Situation bezüglich der mathematischen Voraussetzungen, die Sie benötigen, komplex (und das nicht nur im wahrsten Sinne des Wortes, da wir in $\mathbb{C}$ arbeiten werden).

Mein Wunsch ist es, nur Schulwissen vorauszusetzen. Doch hier kommt es allein schon deswegen zu Konflikten, da aufgrund des föderalen Systems das Niveau der Mathematikkenntnisse auf Bundesebene nicht einheitlich ist. Aber das größte Problem ist, dass ich gestehen muss, dass auch im besten Fall das Schulwissen (und insbesondere die pädagogische Vermittlung des

wissenschaftlichen Arbeitens in der Schule) nicht ausreicht, um alle Details in diesem Buch auf Anhieb vollständig zu verstehen. Daher werden wir in der Vorbereitung einiges aufarbeiten, auch wenn dies nur sehr kurz ausfallen kann, was allerdings für die Beweise und Herleitungen in diesem Buch genügen sollte. Doch bevor Sie sich weiterführender Literatur widmen, empfehle ich Ihnen, diese mathematischen Grundlagen auch tiefergehend zu studieren (falls es notwendig ist).

Es muss allerdings auch erwähnt werden, dass es für viele der genannten Ziele nicht absolut notwendig ist, jeden Beweis zu verstehen. Vielleicht ergibt es für Sie Sinn, das Buch in einem ersten Durchgang zu lesen, ohne die mathematische Tiefe auszureizen. Anschließend können Sie dann diese wichtigen mathematischen Fragestellungen nachholen.

Eine letzte Anmerkung hierzu: Wenn Sie auf universitärer Ebene (sei es zum Beispiel im Informatik- oder im Physik-Studium) einen Kurs zur *Linearen Algebra* gehört und erste Erfahrungen in der *Wahrscheinlichkeitstheorie* gesammelt haben, dann sind Sie auf mathematischer Ebene für dieses Buch bestens gerüstet. Sie müssen sich in diesem Fall nur noch an die spezielle Notation – die *Dirac-Notation* – der Quantenmechanik gewöhnen.

Vorwissen aus der Physik

Wie in der Auflistung der Ziele angedeutet, spielt der physikalische Aspekt der Quantenmechanik nur eine ergänzende Rolle in diesem Buch. Die meisten oben formulierten Ziele sind auch ohne das vollständige Verständnis dieses Kapitels erreichbar. Es ist nur wichtig, dass Sie wenigstens ein oberflächliches Verständnis für dessen Inhalte erhalten. Daher werde ich auf eine Aufbereitung des physikalischen Vorwissens verzichten.

Vorwissen aus der Informatik

Der Informatik-Aspekt spielt in diesem Buch insbesondere bei den Codebeispielen eine wichtige Rolle. Da diese Codebeispiele aber als ergänzende Hilfsmittel zum Verständnis der abstrakten Theorie herangezogen werden, setze ich bei der Informatik mehr Vorwissen bei Ihnen voraus. So gehe ich davon aus, dass Sie genug Kenntnisse in Linux und Python besitzen, um Qiskit und die Codebeispiele einrichten zu können. Ich bin aber überzeugt, dass Sie auch ohne derartige Kenntnisse – vielleicht mit ein bisschen mehr zeitlichem Aufwand – die Codebeispiele nachvollziehen können, sofern Sie bereits erste Erfahrungen im Programmieren gesammelt haben.

Wie Sie die Entwicklungsumgebung Qiskit einrichten, zeige ich konkret im nächsten Abschnitt. Dort finden Sie auch Verweise auf die ausführliche Dokumentation, in der Sie die Details nachschlagen können.

1.6 Qiskit – eine Entwicklungsumgebung für Quantencomputer

Ein wesentliches Merkmal meiner Herangehensweise in diesem Buch ist, dass wir stets praktische Beispiele zur Hand nehmen werden, damit Sie sich die mitunter abstrakte Theorie besser visualisieren und einprägen können. Doch wie kann man sich praktisch in der Quanteninformatik austoben, wenn es – wie gesagt – noch gar keine richtigen Quantencomputer gibt?

Die Antwort ist, dass wir einen Quantencomputer so gut es geht simulieren müssen! Aufgrund der in der Theorie möglichen deutlichen Überlegenheit von Quantencomputern ist es klar, dass so eine Simulation schnell an ihre Grenzen kommen wird. Wir werden uns daher mit Rechen- und Programmierbeispiele in kleinen Maßstäben begnügen müssen. Aber ich versichere Ihnen, dass bereits diese kleineren Beispiele Sie wesentlich dabei unterstützen werden, das Thema Quantencomputing zu verstehen.

Die Plattform, die wir hierzu nutzen werden, ist IBMs Entwicklungsumgebung *Qiskit*, die auf der Programmiersprache *Python* aufgebaut ist. Es handelt sich hierbei um eine Open-Source-Umgebung; sie ist für Sie somit frei und einfach zugänglich. Tatsächlich gäbe es auch die Möglichkeit, die Berechnungen auf einem echten Quantencomputer von IBM durchführen zu lassen, was aber selbstverständlich Kosten verursacht und für unseren Einstieg nicht notwendig ist. Wir möchten uns aber auf die Simulation eines Quantencomputers beschränken. Dies sollte für unsere Zwecke reichen. Falls Sie wissen möchten, wie Sie Zugang zu Rechenzeit am Quantencomputer erhalten, verweise ich Sie auf die jeweiligen Webseiten von IBM:

https://www.ibm.com/quantum/pricing

Qiskit in einem konkreten Szenario

Wie bereits in Abschnitt 1.5.3 erwähnt, gehe ich davon aus, dass Sie bereits Vorkenntnisse im Programmieren haben und keine ausführliche Einführung in Python benötigen. Daher werde ich nur rudimentär und in einem speziellen Szenario erklären, wie Sie Qiskit so einrichten, dass Sie die Beispiele in den folgenden Kapiteln folgen können. Ich gehe dabei davon aus, dass Sie in einer Linux-Umgebung arbeiten und bereits Python 3 installiert haben. Ich empfehle Ihnen für unsere Zwecke die Plattform *Jupyter Notebook*, die mittlerweile sehr bekannt und beliebt ist. Alle nötigen Informationen finden Sie unter *https://jupyter.org*.

Um in diesem konkreten Szenario Qiskit einzurichten und anschließend die für uns nötigen Python-Pakete zu installieren, geben Sie bitte folgende Befehle in das Terminal ein:

```
sudo apt install python3, python3-pip, jupyter-core, jupyter-notebook

pip install qiskit==0.46
pip install qiskit_aer
pip install matplotlibq
pip install pylatexenc
pip install galois
```

Listing 1.1 Die benötigten Pakete, wenn Sie Ubuntu nutzen.

Wenn Sie eine andere Linux-Distribution nutzen, müssen Sie den Befehl `apt` entsprechend austauschen und schauen, ob die Pakete einen leicht anderen Namen tragen. Beachten Sie außerdem, dass ich Sie bewusst die Version 0.46 von Qiskit installieren ließ, da die Codebeispiele in diesem Buch auf dieser Version basieren und wir nur so sichergehen können, dass es nicht zu Problemen mit der Kompatibilität kommt. Damit sind Sie nun gerüstet, um die kommenden Codebeispiele auszuprobieren.

Hello World und Tutorials

Wenn Sie erst einmal mit Qiskit warm werden wollen, sollten Sie die Dokumentation studieren. Sie ist sehr ausführlich und wird sogar mit Videos unterstützt. Wie jeder gute Einstieg startet auch dieses Tutorial mit einem `Hello World`: *https://docs.quantum.ibm.com/guides/hello-world*. Unser erstes Codebeispiel – d. h. unser `Hello World` – finden Sie in Abschnitt 4.2.2.

Außerdem stellt IBM einige gute und ausführliche Tutorials bereit, die Sie als Ergänzung zu diesem Buch durcharbeiten können. Den Katalog der Übungen finden Sie unter:

https://learning.quantum.ibm.com/catalog/tutorials

1.7 Plan-Check-Do-Act

Aus jeder Anstrengung, in die Lebenszeit, Energie, Leidenschaft und Freude eingeflossen sind, erwächst der Wunsch, dass die Arbeit Früchte trägt. Es ist mir deshalb persönlich wichtig, dass Ihnen als Leserin oder Leser dieses Buch Freude und Lernerfolge ermöglicht. Daher freue ich mich über jede – und wirklich jede – Art von Rückmeldung, Kritik, Hinweis und Verbesserungsvorschlag, um die Qualität dieses Buchs zu verbessern.

Diese Qualitätssteigerung gelingt am besten über den direkten Austausch. Das hat die Erfahrung in der Lehre stets bestätigt und ist auch Teil des Plan-Check-Do-Act-Zyklus eines guten Qualitätsmanagements (wie Sie es vielleicht aus der ISO 9001 kennen).

Bitte nutzen Sie hierfür die E-Mail-Adresse:

quantencomputing@rheinwerk-verlag.de

Ich freue mich auf Ihre Nachrichten.

1.8 Ausblick auf die Inhalte dieses Buchs

In diesem Abschnitt möchte ich einen kurzen Überblick über dieses Buch geben. Es besteht aus fünf Teilen: **Vorbereitung**, **Elemente der Quantenmechanik**, **Quantenschaltkreismodell**, **Algorithmen** und **Nachworte**.

Nach der **Einleitung** werden wir in Teil I, der **Vorbereitung**, unsere Kenntnisse auf denselben Stand bringen. Ich liste hierzu einige Begriffe und Konzepte auf, die für den weiteren Verlauf dieses Buchs vorausgesetzt werden. Dazu zählen zum einen einige *mathematische Grundlagen* sowie zum anderen ein kurzer Blick auf einige wichtige *Experimente aus der Quantenphysik* in Kapitel 3, aus denen die Postulate der Quantenmechanik hervorgegangen sind.

Die ersten wesentlichen Schritte dieses Buchs machen wir in Teil II, **Elemente der Quantenmechanik**. Hier werden Sie die für uns wichtigen mathematischen Grundlagen aus der *Quantenmechanik* erarbeiten. Die großen Meilensteine in Kapitel 4 sind die einzelnen *Postulate der Quantenmechanik*. In Kapitel 5 schauen wir uns direkte Konsequenzen aus diesen Postulaten an. Insbesondere werden Sie hierbei das Phänomen der *Verschränkung* kennenlernen.

In Teil III, **Quantenschaltkreismodell**, sind Sie nun bereit, in die *Quanteninformatik* einzusteigen. Ich lege den Fokus auf das Berechnungsmodell des *Quantenschaltkreismodells*. Hierzu führe ich in Kapitel 6 *(Multi-)Qubits* und in Kapitel 7 *Gatter* ein, um diese dann in Kapitel 8 zu *Quantenschaltkreisen* zusammenzuführen.

In Teil IV ernten Sie den Lohn Ihrer Arbeit aus den vorhergehenden Teilen und lernen einige wichtige **Quantenalgorithmen** kennen:

- den *Deutsch-Jozsa-Algorithmus* (Kapitel 9),
- den *Simon-Algorithmus* (Kapitel 10),
- den *Grover-Algorithmus* (Kapitel 11) und
- den *Shor-Algorithmus* (Kapitel 12).

In den **Nachworten**, Teil V, finden Sie einen kurzen Exkurs in ein damit zusammenhängendes, sehr spannendes Thema: die *Post-Quanten-Kryptografie*.

Des Weiteren werfen wir auch noch einen Blick auf weitere Themen, die ich Ihnen als nächste Schritte ans Herz lege.

Natürlich habe ich die Reihenfolge im Buch bewusst gewählt und meine erste Empfehlung ist daher, dass Sie vorne beginnen und das Buch sukzessive durcharbeiten. Aber Sie können auch – je nach Vorkenntnissen und ihrem Fokus – einige Kapitel oder Abschnitte nur sichten oder gar überspringen. Beispielsweise können Sie einzelne Abschnitte aus Teil I ruhigen Gewissens nur überfliegen, falls Sie bereits über grundlegende Kenntnisse in der Linearen Algebra, der Wahrscheinlichkeitstheorie, im Programmieren oder in der Quantenphysik verfügen.

Kapitel 4 und Kapitel 5 sind die Grundpfeiler dieses Buchs und daher würde ich Ihnen nicht empfehlen, diese zu überspringen. Ebenso empfehle ich Ihnen, Teil III sorgfältig durchzuarbeiten, auch wenn Sie die abgeleiteten Konzepte der Quanteninformatik vielleicht schon aus den Kapiteln 4 und 5 erahnen können.

Die Algorithmen in Teil IV können unabhängig voneinander gelesen werden, auch wenn natürlich Parallelen und Zusammenhänge zwischen diesen Algorithmen existieren.

Die Inhalte von Teil V stehen nicht in direktem Zusammenhang mit den anderen Teilen und können unabhängig von diesen gelesen werden.

4.4.2 Beabsichtigte Manipulation des Zustands durch unitäre Abbildungen

Nun haben Sie gelernt, welche Gesetze für die Evolution in einem abgeschlossenen Quantensystem gelten (bei dem der Hilbertraum gegeben ist durch $\mathbb{C}^k$). Häufig beabsichtigt man aber vielmehr, Quantensysteme bewusst und nach einem bestimmten Schema zu manipulieren. Beispielsweise kann man in der Quanteninformatik nur so gezielt Quantenalgorithmen implementieren. Daher ist die eigentliche Frage, der wir uns stellen müssen:

⋆ Wie kann man Zustände in einem Quantensystem bewusst verändern oder in eine gewünschte Richtung lenken?

Es gibt natürlich verschiedene Ansätze, die Frage ⋆ zu beantworten. An dieser Stelle möchte ich mich auf den Ansatz konzentrieren, der die wesentliche Grundlage für die Quanteninformatik[3] bildet.

Zur Frage (⋆) Sei $|\psi\rangle \in \mathbb{C}^k$ der Zustand des Quantensystems zur Zeit $t_0 = 0$. Wir beabsichtigen, $|\psi\rangle$ so zu manipulieren, dass ein neuer gewünschter Zustand angenommen wird. Dieser neue gewünschte Zustand könnte beispielsweise die sogenannte Quanten-Fouriertransformation von $|\psi\rangle$ (siehe Abschnitt 12.2.2) sein.[4] Wie können wir dieses Ziel erreichen? Hierzu erinnern wir uns an Lemma 4.4.2: Nach einer Zeit t wird der Zustand $|\psi_t\rangle = \exp(-i\mathrm{H}t)\ |\psi\rangle$ angenommen, wobei H ein Hamilton-Operator ist, der die Einflüsse beschreibt, die auf das Quantensystem in diesem Zeitraum wirken. Eine Strategie zur Beantwortung von Frage ⋆ wäre nun:

Wähle einen Hamilton-Operator H, bei dem wir wissen, dass $\exp(-i\mathrm{H}t)\ |\psi\rangle$ einen gewünschten neuen Zustand darstellt (für ein geeignetes t).

So einen geeigneten Hamilton-Operator zu finden und diesen anschließend physikalisch zu implementieren ist (sowohl mathematisch als auch physi-

3 Genauer gesagt basiert das Berechnungsmodell *Quantenschaltkreismodell* der Quanteninformatik auf diesen Ansatz. Dieses Modell wird in Definition 8.1.1 eingeführt und liegt – wie bereits erwähnt – diesem Buch als Basis zugrunde. Da ich dieses Modell noch nicht rigoros eingeführt habe und mit unbekannten Begriffen sparsam umgehen möchte, spreche ich in diesem Abschnitt von »Quanteninformatik«, aber meinen eigentlich genauer das »Quantenschaltkreismodell«.

4 Die Vorgabe, dass ein einziger neuer gewünschter Zustand erreicht werden soll, kann auch etwas abschwächt werden: Manipuliere $|\psi\rangle$ so, dass der neue Zustand ein Element einer Zielmenge $Z \subset \mathbb{C}^k$ ist. Bei dieser Zielmenge Z könnte es sich beispielsweise um die Menge aller Zustände handeln, die sich in der Nähe (bezüglich einer gewissen Metrik) einer Lösung eines linearen Gleichungssystems befinden (siehe der sogenannte *HHL-Algorithmus* in [HHL09]).

kalisch) eine große Herausforderung. In der Quanteninformatik wählt man daher eine andere Strategie. Diese basiert auf der folgenden Beobachtung:

Lemma 4.4.3 Sei $K \in \mathcal{L}_{sa}(\mathbb{C}^k)$. Dann ist $U = \exp(-iK)$ eine unitäre Abbildung, d.h., $U \in \mathcal{U}(\mathbb{C}^k)$.

Umgekehrt sei $U \in \mathcal{U}(\mathbb{C}^k)$. Dann gibt es ein $K \in \mathcal{L}_{sa}(\mathbb{C}^k)$, sodass $U = \exp(-iK)$.

Beweis: Der Beweis dieses Lemmas erfolgt in Übung 4.5. □

Aus dem vorangehenden Lemma lernen Sie, dass es eine 1-zu-1-Beziehung zwischen unitären und selbstadjungierten Abbildungen gibt. Das bedeutet, dass wir, anstatt nach geeigneten $\mathrm{H} \in \mathcal{L}_{sa}(\mathbb{C}^k)$ und t zu suchen, auch nach einem geeigneten $U \in \mathcal{U}(\mathbb{C}^k)$ suchen können. Diese Objekte sind miteinander verbunden durch $U = \exp(-it\mathrm{H})$.

Diese Beobachtung liefert uns folgende *Antwort auf Frage (⋆) für die Quanteninformatik*:

Suche nach einer unitären Abbildung U, sodass $U|\psi\rangle$ einen gewünschten neuen Zustand annimmt.

Diese Strategie bringt einige hilfreiche Vorteile mit sich: Es ist mathematisch leichter, direkt nach einer passenden unitären Abbildung zu suchen statt nach einer selbstadjungierten Abbildung, die erst nach Anwendung der Exponentialabbildung passend sein soll für die gesuchte Transformation des aktuellen Zustands. Des Weiteren ist es ein bekanntes Resultat aus der Linearen Algebra, dass sich (in $\mathbb{C}^k$) alle unitären Abbildungen als Verknüpfung von einfacheren Matrizen darstellen lassen (siehe Abschnitt 7.6). Somit reicht es aus, diese einfacheren Matrizen zu implementieren, da sich alle anderen unitären Matrizen auf diesen Fall reduzieren lassen. Ein weiterer Vorteil ist die Tatsache, dass die Erhaltung der Normiertheit der Quantenzustände direkt sichtbar ist. Diese Tatsache ist eine direkte Konsequenzen aus der Definition von unitären Abbildungen (siehe Abschnitt 2.4).

Aus der Beantwortung von Frage $\star$ haben wir Folgendes gelernt: Beabsichtigte Änderungen des Zustands in der Quanteninformatik lassen sich am besten durch unitäre Abbildungen umsetzen. Um dies in der Praxis auch wirklich anzuwenden, müssen Sie aber noch etwas Wichtiges beachten: Diese unitären Abbildungen müssen so gewählt sein, dass wir sie auch physikalisch in unserem Quantensystem realisieren können. Das heißt, wir müssen wis-

sen, wie wir das abgeschlossene System *von außen* manipulieren können, um als Resultat die Anwendung dieser unitären Abbildung auf den Zustand des Quantensystems zu erhalten. Hier kann man darauf zurückgreifen, dass sich – wie bereits erwähnt – alle infrage kommenden unitären Abbildungen durch einen gewissen Satz an unitären Matrizen, die sogenannten *Universalgatter der Quanteninformatik*, darstellen lassen (siehe Abschnitt 7.6). Somit ist es ausreichend, wenn Sie wissen, wie man die Universalgatter in der Praxis implementieren muss. Hierauf gehe ich in Abschnitt 7.6 genauer ein.

Sie wissen nun, wie wir in der Quanteninformatik Manipulationen des Quantenzustands tätigen werden. Eine direkte Folgefrage lautet: *Wie können wir dieses Wissen nutzen, um Quantenalgorithmen zu implementieren?* Dieser Frage werden wir in Teil III genauer nachgehen. Doch bereits an dieser Stelle können Sie erahnen, was die Lösung sein wird: *Wende sukzessive geeignete unitäre Abbildungen an* – gemäß dem Quantenalgorithmus –, um den angestrebten Quantenzustand zu erhalten.

Die Erkenntnisse dieses Abschnitts liefern uns folgendes Postulat. Gemäß unserer Strategie in diesem Kapitel formulieren wir dieses Postulat für allgemeine (endlichdimensionale) Hilberträume, obwohl wir in der vorangehenden Motivation nur den Fall $\mathbb{C}^k$ betrachtet haben. Natürlich lässt sich das Vorangehende auch leicht auf allgemeinere Hilberträume abstrahieren.

Postulat II.2 Sei $\mathscr{H}$ ein Hilbertraum. In einem abgeschlossenen System sei

- der Zustand zur Zeit $t_0 \geq 0$ beschrieben durch $|\psi_0\rangle \in \mathscr{H}$,
- der Zustand zur Zeit $t_1 > t_0$ beschrieben durch $|\psi_1\rangle \in \mathscr{H}$.

Dann gibt es eine unitäre Abbildung $U \in \mathcal{U}(\mathscr{H})$, sodass

$$|\psi_1\rangle = U\,|\psi_0\rangle\,. \tag{4.4.3}$$

Bemerkung 4.4.4 Eine weitere interessante Eigenschaft der Quantenmechanik lässt sich aus Postulat II.2 (oder aber auch aus Lemma 4.4.2) ablesen: *die Reversibilität in abgeschlossenen Systemen*. Dies ist eine Folgerung aus der Tatsache, dass unitäre Abbildungen stets invertierbar sind (oder aus der Tatsache, dass die Exponentialabbildung in Lemma 4.4.2 invertierbar ist). Somit müssen Manipulationen des Zustands eines Quantensystem in einen neuen Zustand immer umkehrbar sein.

Insbesondere gilt diese Eigenschaft dann auch in der Quanteninformatik. Tatsächlich ist die Umkehrung einer Teilroutine eines Quantenalgorithmus ein übliches Vorgehen, was man auch als *Uncomputation* bezeichnet. In der Regel beabsichtigt man durch die Uncomputation, ein Hilfsregister – nachdem es seine

Pflichten erledigt hat – wieder in den Grundzustand zurückzusetzen und zu verwerfen.

Vielleicht fragen Sie sich, wieso man nicht einfach so das Register verwerfen kann und vorher erst die Uncomputation erledigen muss. Dies liegt daran, dass nach einigen Berechnungen oftmals die Register mittels des Phänomens der *Verschränkung* gekoppelt sind. Daher muss man erst die Kopplung/Verschränkung rückgängig machen und erst dann kann man das Hilfsregister verwerfen. Das Phänomen der Verschränkung lernen Sie in Abschnitt 5.3 kennen. Und ein Beispiel für eine Uncomputation findet sich beispielsweise beim sogenannten *HHL*-Algorithmus von Harrow, Hassidim und Lloyd (siehe [HHL09]).

In der klassischen Informatik ist Reversibilität nur in speziellen Situationen vorhanden. Denn die üblichen Gatter der klassischen Informatik (wie beispielsweise das NAND- oder XOR-Gatter) sind *nicht reversibel*, da diese zwei Eingabewerte haben, aber nur einen Ausgabewert.

4.4.3 Zwei wichtige Beispiele

Die folgenden Beispiele schließen an den letzten Abschnitt an. Hierbei werden Sie auch unsere nächsten Codebeispiele sehen. In diesem Abschnitt gilt, dass der Hilbertraum gegeben ist durch

$$\mathscr{H} = \mathcal{H} = \mathbb{C}^2,$$

da wir auf Beispiele zurückgreifen, die für die Quanteninformatik (und somit auch für dieses Buch) sehr relevant sind.

Die Hadamard-Matrix

Das wohl typischste Beispiel für eine Manipulation des Zustands in der Quanteninformatik ist die sogenannte *Hadamard-Matrix*.

Definition 4.4.5 Die **Hadamard-Matrix** $H \in \mathbb{C}^{2,2}$ ist definiert durch:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{4.4.4}$$

Bevor wir uns anschauen, wie genau die Hadamard-Matrix in Qiskit abgebildet wird, liste ich im folgenden Lemma ein paar Eigenschaften dieses Operators auf.

Lemma 4.4.6 Es gilt:

$$H^* = H \quad \text{und} \quad H \cdot H = \mathrm{Id},$$

d. h., die Hadamard-Matrix ist sowohl selbstadjungiert als auch unitär. Des Weiteren gilt, dass

1. $H(|0\rangle) = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
2. $H(|1\rangle) = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Beweis: Der Beweis dieses Lemmas wird in Übung 4.6 geführt. □

Wie bereits angedeutet, ist die Hadamard-Matrix eine der wichtigsten Transformationen in der Quanteninformatik. Daher widmen wir unser nächstes Codebeispiel in diesem Buch dieser Matrix.

Codebeispiel 4.4.7 Das Codebeispiel aus Listing 4.2 ist ähnlich aufgebaut wie das Codebeispiel aus Listing 4.1. Wir verzichten aber hier auf eine individuelle Präparation des Qubits und nutzen, dass per Default das Qubit stets auf den Wert $|0\rangle$ präpariert wird. Wichtig ist, dass wir ergänzend nach der Präparation des Qubits die Hadamard-Matrix auf dieses Qubit anwenden. Dies erfolgt durch den Befehl `h(0)`. Der Eingabewert `0` signalisiert, bei welchem Qubit die Hadamard-Matrix angewendet wird. In diesem Fall ist das wenig überraschend, da das Quantensystem nur ein Qubit enthält.

Letztlich müssen wir noch zur Ausgabe des Zustands des Quantensystems den Befehl `Statevector(circ)` einbauen, um den Zustandsvektor des Quantensystems extrahieren zu können.

In Listing 4.2 sehen Sie den zugehörigen Code.

```
from qiskit import QuantumCircuit
from qiskit.quantum_info import Statevector

circ = QuantumCircuit(1)

# Wende die Hadamard-Matrix an
circ.h(0)

Statevector(circ)
```

Listing 4.2 Anwendung der Hadamard-Matrix auf ein Qubit

Dies führt zu folgender Ausgabe:

```
Statevector([0.70710678+0.j, 0.70710678+0.j],
                    dims=(2,))
```

Listing 4.3 Die Ausgabe

Wir möchten noch prüfen, ob die Ausgabe unseren Erwartungen entspricht. Gemäß Lemma 4.4.6 erwarten wir den Vektor

$$H(|0\rangle) = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

als Ausgabe. Unter Berücksichtigung der Tatsache, dass 0.70710678 eine Approximation von $1/\sqrt{2}$ ist, sehen wir, dass das Codebeispiel unsere Erwartungen bestätigt.

Wir schließen dieses Beispiel noch mit der folgenden Frage ab: *Wieso ist die Hadamard-Matrix eigentlich so wichtig?*

Hier gibt es natürlich einige gute Gründe. Die folgenden drei Beobachtungen machen ihre Bedeutung bereits sehr deutlich:

- H erzeugt Superposition! Dies ist eine direkte Folgerung aus den Identitäten (i) und (ii) in Lemma 4.4.6. Beispielsweise fangen wir in (i) mit dem Zustand $|0\rangle$ an und nach Anwendung von H wird eine Superposition von $|0\rangle$ und $|1\rangle$ erzeugt (siehe Abschnitt 4.3). Wie ich bereits nach Gleichung (4.2.9) erwähnt habe, spricht man hier sogar von einer *gleichförmigen Superposition*.
- Über die Bedeutung der Basis $\{|+\rangle, |-\rangle\}$ haben wir bereits nach deren Definition in Gleichung (4.2.9) gesprochen. Aufgrund der Identitäten (i) und (ii) sehen Sie, dass H zwischen der Rechenbasis und dieser Basis abbildet.
- In vielen wichtigen Quantenalgorithmen spielt H eine wichtige Rolle, beispielsweise beim *Deutsch-Jozsa-Algorithmus* (siehe Kapitel 9) oder bei der *Quanten-Fouriertransformation* (siehe Abschnitt 12.2.2).

Die Pauli-Matrizen

Die Beispiele, die Sie nun kennenlernen werden, sind die sogenannten *Pauli-Matrizen*, die in der Quanteninformatik eine sehr bedeutende Rolle einnehmen und die uns deshalb in diesem Buch immer wieder begegnen werden.

Definition 4.4.8 Die Definitionen der Pauli-Matrizen lauten:
Die **Pauli-X-Matrix** $\sigma_x \in \mathbb{C}^{2,2}$ ist definiert durch

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

Die **Pauli-Y-Matrix** $\sigma_y \in \mathbb{C}^{2,2}$ ist definiert durch

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}.$$

Die **Pauli-Z-Matrix** $\sigma_z \in \mathbb{C}^{2,2}$ ist definiert durch

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

Im Folgenden möchte ich einige nützliche Eigenschaften dieser Pauli-Matrizen auflisten.

Lemma 4.4.9 Seien $\{\sigma_j\}_{j=x,y,z} \subset \mathbb{C}^{2,2}$ die Pauli-Matrizen. Dann gilt für $j \in \{x, y, z\}$:

$$\sigma_j^* = \sigma_j \quad \text{und} \quad \sigma_j^2 = \sigma_j \cdot \sigma_j = \mathrm{Id},$$

d. h. die Pauli-Matrizen sind sowohl selbstadjungiert als auch unitär.

Für die **Pauli-X-Matrix** σ_x gilt, dass

$$\sigma_x(|0\rangle) = |1\rangle \quad \text{und} \quad \sigma_x(|1\rangle) = |0\rangle.$$

Für die **Pauli-Y-Matrix** σ_y gilt, dass

$$\sigma_y(|0\rangle) = i\,|1\rangle \quad \text{und} \quad \sigma_y(|1\rangle) = -i\,|0\rangle.$$

Für die **Pauli-Z-Matrix** σ_z gilt, dass

$$\sigma_z(|0\rangle) = |0\rangle \quad \text{und} \quad \sigma_z(|1\rangle) = -\,|1\rangle.$$

Beweis: Der Beweis dieses Lemmas erfolgt durch direktes Nachrechnen der behaupteten Identitäten. Insbesondere für die letzten Identitäten ist das sehr einfach, da die Multiplikation einer Matrix mit $|0\rangle$ (bzw. mit $|1\rangle$) die erste Spalte (bzw. die zweite Spalte) dieser Matrix ausgibt. □

Sie sehen also, dass die Pauli-Matrizen (insbesondere) unitäre Matrizen sind, sodass sie zur Manipulation von Qubits infrage kommen. Und hierbei spielen sie auch eine wichtige Rolle, die wir basierend auf Lemma 4.4.9 und der folgenden Argumentation ableiten werden: Erinnern Sie sich an die in Abschnitt 4.2.2 erwähnte informationstheoretische Interpretation der Rechenbasis $\{|0\rangle, |1\rangle\}$. Kombiniert mit den Identitäten aus dem vorangehenden Lemma erkennen Sie folglich, wieso die Pauli-Matrizen auch für die Manipulation von Qubits wichtig sind, da mit diesen *Bitflips* und sogenannte

Phasenflips modelliert werden können. Ich werde in Beispiel 5.1.5 auf die Bezeichnung *Phasenflip* eingehen, weil Sie bis dahin auch gelernt haben, was genau eine *Phase* ist.

Dies unterstreicht schon mal die Bedeutung der Pauli-Matrizen. Es gibt aber noch zwei weitere Punkte, die mitunter die Wichtigkeit dieser Matrizen begründen:

- Auf mathematischer Ebene spielen diese Matrizen eine wichtige Rolle, da sie gemeinsam mit der Identität eine Basis von $\mathcal{L}_{sa}(\mathcal{H})$, dem Raum der hermiteschen Matrizen auf $\mathcal{H}$, bilden.
- Die Pauli-Matrizen sind sehr wichtig beim Thema *Messung*, wie Sie in Abschnitt 4.5.1 sehen werden.

Aufgrund der Bedeutung der Pauli-Matrizen widmen wir diesen unsere nächsten Codebeispiele. Wir möchten nun zwei Qiskit-Codes schreiben: einen zur Implementierung der Pauli-X-Matrix, einen zur Implementierung der Pauli-Z-Matrix. Wir gehen dabei ähnlich vor wie bei der Hadamard-Matrix, jedoch mit einem wesentlichen Unterschied: Wir werden jeweils andere Wege nutzen, um das Qubit mit einem speziellen Zustand zu präparieren. Hierdurch können Sie nebenbei auch diese neuen Befehle kennenlernen.

Codebeispiel 4.4.10 Für die Pauli-X-Matrix erhalten wir den Code in Listing 4.4 mit entsprechender Ausgabe. Der Befehl `x` steht hier für die Anwendung der Pauli-X-Matrix. Für die Präparation des Qubits nutzen wir den Befehl `Statevector.from_int(1,2)`, wobei hier der erste Eingabeparameter eine ganze Zahl angibt, die den gewünschten Zustand beschreibt. In diesem Fall ist das die 1, womit wir den Zustand $|1\rangle$ präparieren. Der zweite Eingabeparameter gibt die Dimension des zugehörigen Hilbertraums an.

```
# Importiere die benötigten Pakete
from qiskit import QuantumCircuit
from qiskit.quantum_info import Statevector

# Initialisiere ein Qubit
circ = QuantumCircuit(1)

# Wende die Pauli-X-Matrix an
circ.x(0)

# Formuliere den Zustandsvektor, der zu präparieren ist
statevector = Statevector.from_int(1,2)
```

```
# Setze den initial präparierten Zustand des Quantensystems
# gleich diesem gewünschten Zustandsvektor und
# initiiere danach die gewünschten Manipulationen
statevector = statevector.evolve(circ)

# Gib den Zustandsvektor aus
print(statevector)
```

Listing 4.4 Anwendung der Pauli-X-Matrix auf ein Qubit

Die Ausgabe sieht so aus:

```
Statevector([1.+0.j, 0.+0.j],
                  dims=(2,))
```

Listing 4.5 Die Ausgabe

Auch hier bestätigt die Ausgabe unsere theoretischen Erfahrungen. In diesem Fall erwarten wir gemäß Lemma 4.4.9, dass es sich bei der Pauli-X-Matrix um eine Bit-Flip-Matrix handelt.

Codebeispiel 4.4.11 Zur Anwendung der Pauli-Z-Matrix gehen wir ähnlich vor. Wir möchten nur gerne die Änderung des Vorzeichens beim Koeffizienten vor $|1\rangle$ nach der Anwendung von σ_z sichtbar machen. Daher möchten wir nun das Qubit mit der gleichförmigen Superposition, $|+\rangle$, initialisieren. Dies gelingt uns beispielsweise durch den Befehl `Statevector.from_label('+')`.

Diesen Befehl kennen Sie aber schon. Daher möchte ich einen anderen Weg einschlagen: Wir greifen das Quantensystem aus Codebeispiel 4.4.7 auf, wo wir bereits den Zustand $H(|0\rangle) = |+\rangle$ präpariert haben. Wir können dann den Zustand aus diesem Quantensystem als initial präparierten Zustand des neuen Quantensystems nutzen, indem wir den Befehl `Statevector.from_instruction(circ)` anwenden. Hier bezeichnet `circ` das Quantensystem aus Codebeispiel 4.4.7. Den zugehörigen Code sehen Sie in Listing 4.6, wobei der Befehl `z` für die Anwendung der Pauli-Z-Matrix steht.

```
from qiskit import QuantumCircuit
from qiskit.quantum_info import Statevector
newcirc = QuantumCircuit(1)

# Wende die Pauli-Z-Matrix an
newcirc.z(0)
# Erhalte den Zustandsvektor, der in circ präpariert wurde
statevector = Statevector.from_instruction(circ)
```

```
# Setze den initial präparierten Zustand des Quantensystems
# gleich diesem gewünschten Zustandsvektor
# und initiiere danach die gewünschten Manipulationen

statevector = statevector.evolve(newcirc)

# Gib den Zustandsvektor aus
print(statevector)
```

Listing 4.6 Anwendung der Pauli-Z-Matrix auf ein Qubit

Sie erhalten dann folgende Ausgabe:

```
Statevector([ 0.70710678+0.j, -0.70710678+0.j],
                  dims=(2,))
```

Listing 4.7 Die Ausgabe

Aufgrund der Tatsache, dass

$$\sigma_z |+\rangle = \sigma_z \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) = \left(\frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle \right),$$

können Sie auch hier die Ausgabe des Codes theoretisch nachvollziehen.

4.4.4 Unbeabsichtigte Änderungen des Zustands

In den letzten drei Abschnitten hatten wir stets die Annahme des *abgeschlossenen Systems* vorausgesetzt. Mit dieser Annahme konnten wir bereits erfolgreich zwei Postulate erarbeiten und sogar erste Quantenalgorithmen aufschreiben. Doch wie bewährt sich diese Annahme in der Praxis?

Wäre die Antwort auf diese Frage positiv, so würde es wohl in jedem Hochleistungsrechenzentrum einen Bereich geben, in dem ein leistungsstarker Quantencomputer betrieben wird. Dass dies bekanntermaßen nicht der Fall ist, sagt uns, dass diese Annahme auf dem Papier zwar Sinn ergibt, aber in der Realität doch weiterer Bemühungen bedarf.

Tatsächlich sind Quantensysteme stets Einflüssen und Störungen aus der Umgebung (wie zum Beispiel durch kosmische Strahlung) ausgesetzt. Solche Einflüsse haben eine direkte Konsequenz für den Zustand der Quantensysteme und verändern diese. Diese ungewollte Veränderung durch die Einflüsse aus der Umwelt bezeichnet man als *Dekohärenz*.

Es zeigt sich, dass diese äußeren Störeinflüsse in Quantensystemen nur durch erheblichen Mehraufwand vermindert werden können. Daher ist es schwer,

Dieser kurze Exkurs in die Historie zeigt, wie bedeutend, spannend und kontrovers die Auswirkungen des Phänomens diskutiert wurden. Dies sollte uns als Motivation für unsere nächsten Schritte dienen.

5.3.2 Von separablen und verschränkten Quantensystemen

Ziel dieses Abschnitts ist es also, das Phänomen der Verschränkung in der Sprache der Quantenmechanik zu beschreiben. Ich werde versuchen, Sie durch intuitive Fragen und durch einen konkreten Fall (gegeben durch das Setting aus Abschnitt 4.6.4) sukzessive an dieses Thema heranzuführen. Die mathematische Beschreibung dieses faszinierenden Phänomens – eingebettet in den quantenmechanischen Formalismus, den wir bisher erarbeitet haben – stellt den Höhepunkt dieses Unterabschnitts dar.

Das einfachste Setting zur Veranschaulichung

Fangen wir mit dem uns bekannten zusammengesetzten Quantensystem an, bei dem der Hilbertraum gegeben ist durch $\mathcal{H} \otimes \mathcal{H}$ (siehe Abschnitt 4.6.4).

Sie kennen mittlerweile den Baustein der Messung, und Sie wissen auch, dass wir uns wahlweise auch bei der Messung nur auf Teile des Quantensystems beschränken können (siehe Beispiel 4.6.5). Was passiert, wenn wir nur bezüglich des einen Quantensystems messen? Das hängt natürlich vom aktuellen Zustand und von der gewählten Observable ab. Im Folgenden schauen wir uns zwei Fälle an, die auftreten können: *separable* und *nicht-separable* Zustände.

Separable Zustände

Schauen wir uns beispielsweise Zustände der Form $|\psi_0\rangle \otimes |\psi_1\rangle$ an, wobei $|\psi_0\rangle, |\psi_1\rangle \in \mathcal{H}$. Solche Zustände, die man als Tensorprodukt von Zuständen aus den Subsystemen schreiben kann, nennt man *separabel*

Wir möchten nun *nur das erste Quantensystem* bezüglich der Observable σ_z messen. Wie Sie in Beispiel 4.6.5 gesehen haben, gelingt eine Messung nur am ersten Quantensystem bezüglich σ_z, indem man für das Gesamtsystem eine Messung bezüglich der Observablen $\sigma_z \otimes I$ durchführt.

Wie hoch sind nun die Wahrscheinlichkeiten, dass das erste Quantensystem nach der Messung den Zustand $|0\rangle$ (bzw. $|1\rangle$) annimmt? Sei hierzu

$$|\psi_0\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

für gewisse $\alpha_0, \alpha_1 \in \mathbb{C}$ wie in Gleichung (4.2.8). Mittels der Linearität des Tensorprodukts erhalten wir dann, dass

$$\begin{aligned} |\psi_0\rangle \otimes |\psi_1\rangle &= (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \otimes |\psi_1\rangle \\ &= \alpha_0(|0\rangle \otimes |\psi_1\rangle) + \alpha_1(|1\rangle \otimes |\psi_1\rangle). \end{aligned}$$

Zur Berechnung der gesuchten Wahrscheinlichkeiten müssen wir natürlich Lemma 4.5.8 für die Observable $\sigma_z \otimes I$ anwenden. Allerdings müssen wir vorher herausfinden, welche Messergebnisse wir erwarten können. Aus Beispiel 4.6.5 wissen wir, dass

- $|0\rangle \otimes |\psi_1\rangle$ und $|1\rangle \otimes |\psi_1\rangle$ Eigenvektoren sind von $\sigma_z \otimes \mathrm{Id}$ zu den Eigenwerten 1 bzw. -1 und dass
- $|0\rangle \otimes |\psi_1\rangle$ und $|1\rangle \otimes |\psi_1\rangle$ orthogonal zueinander sind.

Dies wiederum impliziert direkt via Lemma 4.5.8 – durch eine ähnliche Berechnung wie in Beispiel 4.5.10 –, dass mit Wahrscheinlichkeit $|\alpha_0|^2$ (bzw. $|\alpha_1|^2$) das Quantensystem nach der Messung den Zustand $|0\rangle \otimes |\psi_1\rangle$ (bzw. $|1\rangle \otimes |\psi_1\rangle$) annimmt.

Insbesondere halten wir aber die folgende wichtige Beobachtung fest: In allen Fällen bleibt das zweite Quantensystem im Zustand $|\psi_1\rangle$. Mit dem zweiten Quantensystem passiert also nichts! Es bleibt *invariant*.

Dies ist ein Umstand, den wir bei *separablen* Zuständen festhalten möchten. Allerdings gilt dies in einem weitaus allgemeineren Setting, wenn man die vorangehenden Berechnungen auf beliebige zusammengesetzte Quantensysteme verallgemeinert. Für diese Verallgemeinerung dieser Aussagen verweise ich auf Kapitel 4 in [Sch19]. Hier halte ich nur das zugehörige Resultat fest:

Proposition 5.3.1 Seien $\mathscr{H}_0, \mathscr{H}_1$ Hilberträume. Sei $|\Psi\rangle \in \mathscr{H}_0 \otimes \mathscr{H}_1$ ein *separabler* Zustand, d. h., es gibt $|\psi_0\rangle \in \mathscr{H}_0$ und $|\psi_1\rangle \in \mathscr{H}_1$, sodass

$$|\Psi\rangle = |\psi_0\rangle \otimes |\psi_1\rangle \,.$$

Wenn der initial präparierte Zustand gegeben ist durch den separablen Zustand $|\Psi\rangle$, dann bleibt bei einer Messung bezüglich des ersten Quantensystems $\mathscr{H}_0$ das zweite Quantensystem $\mathscr{H}_1$ invariant. Genauso verhält es sich auch bei einer Messung bezüglich des zweiten Quantensystems.

Beweis: Dies ist eine Verallgemeinerung der vorangehenden Erkenntnisse, die in Kapitel 4 in [Sch19] (insbesondere in Theorem 4.3) nachgelesen werden kann. Bei der Formulierung wird in dieser Referenz das Konzept von *gemischten Zuständen* benutzt, auf das ich in diesem Buch verzichtet habe. □

Nicht-separable Zustände

Doch was passiert bei *nicht-separablen* Zuständen, d. h. bei Zuständen, die wir nicht als Tensorprodukt von Zuständen aus den einzelnen Hilberträumen definieren können? Auch hier bietet es sich an, explizit ein Beispiel durchzurechnen und daraus Erkenntnisse zu gewinnen.

Betrachten wir eines der Elemente aus der *Bell-Basis*, die wir in Gleichung (4.6.18) kennengelernt haben:

$$|\Psi_+\rangle = \frac{1}{\sqrt{2}}\Big(|01\rangle + |10\rangle\Big) = \frac{1}{\sqrt{2}}\Big(|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle\Big) \tag{5.3.1}$$

In Übung 5.6 zeige ich, dass dieser Zustand tatsächlich nicht-separabel ist.

Was passiert, wenn wir hier bezüglich der Observablen $\sigma_z \otimes I$ messen? Auch hier müssen wir erst die Eigenvektoren von $\sigma_z \otimes \mathrm{Id}$ ermitteln. Eine einfache Berechnung hierzu zeigt, dass

- $|0\rangle \otimes |1\rangle$ und $|1\rangle \otimes |0\rangle$ Eigenvektoren sind von $\sigma_z \otimes \mathrm{Id}$ zu den Eigenwerten 1 bzw. -1 und dass
- $|0\rangle \otimes |1\rangle$ und $|1\rangle \otimes |0\rangle$ orthogonal zueinander sind.

Auch hier impliziert eine direkte Anwendung von Lemma 4.5.8, dass mit Wahrscheinlichkeit 1/2 der Zustand $|01\rangle$ nach der Messung angenommen wird und mit Wahrscheinlichkeit 1/2 der Zustand $|10\rangle$.

Dies ist sehr interessant. Denn jetzt ist es *nicht* so, dass in beiden Fällen der Zustand des zweiten Quantensystems unabhängig ist vom Messausgang des ersten Quantensystems. Vielmehr legt die Messung – erinnern Sie sich daran, dass wir nur bezüglich des ersten Quantensystems gemessen haben (da wir $\sigma_z \otimes I$ als Observable gewählt haben) – bereits das zweite Quantensystem fest, obwohl wir es gar nicht angefasst haben (da wir ja bezüglich des zweiten Quantensystems die Identität I als Observable gewählt haben). Genauer gesagt erhalten wir Folgendes:

- Wenn wir im ersten Quantensystem den Zustand $|0\rangle$ beobachtet haben, wissen wir, dass das zweite Quantensystem nach der Messung den Zustand $|1\rangle$ angenommen hat.
- Wenn wir im ersten Quantensystem den Zustand $|1\rangle$ beobachtet haben, wissen wir, dass das zweite Quantensystem den Zustand $|0\rangle$ angenommen hat.

Wir halten fest, dass bei nicht-separablen Quantensystemen eine Messung des ersten Quantensystems unmittelbare Auswirkungen auf den Zustand des zweiten Quantensystems hat. Dies ist eine Realisation des Phänomens der *Verschränkung*, und nicht-separable Quantensysteme nennt man daher *verschränkte* Quantensysteme.

Interessanterweise erkennt man bereits in diesen formalen Berechnungen, dass bei dieser Wirkung aufeinander bei verschränkten Quantensystemen nirgendwo die Komponente der räumlichen Distanz zwischen diesen Zuständen eine Rolle spielt. Sobald wir in der Lage sind, verschränkte Zustände – wie

beispielsweise den oben analysierten Bell-Basisvektor – zu erzeugen und diese zu halten, können wir diese Wirkung erwarten. Und dies ist mittlerweile beispielsweise über eine Distanz von 230 m – in diesem Fall via sogenannter *Ionenfallen* – möglich (siehe [KGK+23]).

Wie in Proposition 5.3.1 lassen sich die soeben hergeleiteten Erkenntnisse auf allgemeine Hilberträume erweitern. Um diese verallgemeinerten Aussagen rigoros beschreiben zu können, benötigt man allerdings das Konzept der *gemischten Zustände*, worauf ich in diesem Buch verzichte. Dennoch möchte ich in der folgenden Proposition eine intuitive Formulierung dieser verallgemeinerten Aussage festhalten. Sie finden die zugehörigen rigorosen Konzepte und Aussagen in Kapitel 2 und 4 von [Sch19].

Proposition 5.3.2 Seien $\mathscr{H}_0, \mathscr{H}_1$ Hilberträume. Sei $|\Psi\rangle \in \mathscr{H}_0 \otimes \mathscr{H}_1$ ein *nichtseparabler* (oder anders ausgedrückt: *verschränkter*) Zustand, d. h., es gibt *keine* $|\psi_0\rangle \in \mathscr{H}_0$ und $|\psi_1\rangle \in \mathscr{H}_1$, sodass

$$|\Psi\rangle = |\psi_0\rangle \otimes |\psi_1\rangle \,.$$

Sei bei einer Messung des ersten Quantensystems $\mathscr{H}_0$ bezüglich einer Observable $A \in \mathcal{L}_{sa}(\mathscr{H}_0)$ der initial präparierte Zustand, gegeben durch den verschränkten Zustand $|\Psi\rangle$. Dann ist der Zustand des zweiten Quantensystems $\mathscr{H}_1$ nach der Messung abhängig vom Ausgang der Messung des ersten Quantensystems.

Genauso verhält es sich auch bei einer Messung bezüglich des zweiten Quantensystems.

Beweis: Der Beweis einer rigorosen Formulierung dieser Proposition mittels des Konzepts von sogenannten *gemischten Zuständen* finden Sie in Theorem 4.3 in [Sch19]. □

Verschränkung und Quantenüberlegenheit

Es steht außer Frage, dass die Verschränkung ein sehr faszinierendes Phänomen ist. Für uns ist aber die Bedeutung in der Quanteninformatik entscheidend. Dass die Superposition eine wichtige Rolle spielen wird, habe ich in Abschnitt 4.3 schon angedeutet. Doch wie ist es mit der Verschränkung?

Es stellt sich heraus, dass Verschränkung notwendig ist für einen *exponential speed-up*, d. h. dafür, dass die Komplexität der Lösung eines Problems auf einem Quantencomputer exponentiell schneller ist als die entsprechende Komplexität auf einem klassischen Computer. Dies ist ein Resultat von Jozsa und Linden in [JL03]. Intuitiv kann man das so erklären: Wenn man

Qubits miteinander verschränkt, Manipulationen des einen Qubits unmittelbare Auswirkungen auf das andere Qubit haben, wodurch Berechnungen deutlich beschleunigt werden können. Für weitere Details empfehle ich Ihnen das Paper [JL03].

5.4 Das No-Communication-Theorem: Das Problem bei Lieferdiensten, die schneller sind als Licht

Vielleicht fragen Sie sich jetzt, ob man aus dieser spukhaften Fernwirkung nicht auch Kapital schlagen könnte, um Informationen zu übertragen. Unsere jetzige Kommunikationsinfrastruktur basiert auf der Übertragung elektromagnetischer Strahlung, die mit Lichtgeschwindigkeit erfolgt. Die Wirkung der Verschränkung erfolgt instantan, also sogar noch schneller. In diesem Abschnitt sehen Sie aber, dass eine noch schnellere Kommunikation *nicht* möglich ist. Das zugehörige Resultat kennt man in der Literatur als das *No-Communication-Theorem*.

Dieses Resultat ist in einem allgemeinen Setting wie in Proposition 5.3.2 gültig. Doch auch hier möchten wir uns auf eine konkrete Instanz beschränken, und die Argumente nur für diese Instanz durchgehen. Wir werden außerdem nur ein mögliches, naheliegendes Kommunikationsprotokoll betrachten – d.h. eine Option, wie sich zwei Kommunikationsparteien verständigen und austauschen können. Die hier präsentierten Argumente liefern aber trotzdem schon die wesentlichen Beweisschritte, die für das allgemeine Setting nötig sind.

Vereinfachtes Setting

Gegeben seien also zwei Kommunikationsparteien *Alice* und *Bob* (wie sie üblicherweise in der Literatur genannt werden), die einen Bitwert 0 oder 1 mithilfe eines verschränkten Quantensystems miteinander austauschen möchten. Wir nehmen an, dass Alice und Bob sehr weit voneinander entfernt sind und ein verschränktes Zwei-Qubit-Register miteinander teilen, wobei Alice das eine Qubit besitzt und Bob das andere. Außerdem nehmen wir der Einfachheit wegen an, dass dieses Zwei-Qubit-Register folgenden Zustand hat:

$$|\Phi_+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{5.4.1}$$

das Sie als Element der Bell-Basis (siehe Gleichung (4.6.18)) wiedererkennen.

Bevor wir weitermachen, erinnere ich Sie an eine Identität, die Sie in Übung 4.20 bewiesen haben:

$$|\Phi_+\rangle = \frac{1}{\sqrt{2}}(|+\rangle \otimes |+\rangle + |-\rangle \otimes |-\rangle) \tag{5.4.2}$$

Kommunikationsprotokoll

Sobald Alice eine Messung an Ihrem Qubit vornimmt, wird das Qubit von Bob einen dementsprechenden Zustand annehmen (entsprechend unserer Erkenntnisse aus 5.3). Und dieser Zustand, den Bob erhält, ist bestimmt durch die Observable, bezüglich der Alice die Messung durchführt. Dies sehen Sie an zwei Beispielen:

- Wählt Alice die Messung bezüglich σ_z, bedeutet das, dass der Zustand ihres Qubits nach der Messung gegeben ist durch entweder $|0\rangle$ oder $|1\rangle$ (jeweils mit Wahrscheinlichkeit 1/2). Aufgrund der Formel (5.4.1) impliziert dies, dass Bobs Qubit nach der Messung von Alice auch entweder den Zustand $|0\rangle$ oder $|1\rangle$ annimmt.
- Wählt Alice die Messung bezüglich σ_x, bedeutet das, dass der Zustand ihres Qubits nach der Messung gegeben ist durch entweder $|+\rangle$ oder $|-\rangle$ (jeweils mit Wahrscheinlichkeit 1/2). Aufgrund der Formel (5.4.2) impliziert dies, dass Bobs Qubit nach der Messung von Alice auch entweder den Zustand $|+\rangle$ oder $|-\rangle$ annimmt.

Das wäre doch ein guter Ansatz, um zu kommunizieren! Das zugehörige Protokoll lautet wie folgt:

- Wenn Alice die 0 übermitteln möchte, dann wählt sie σ_z als Observable. Danach hat Bobs Qubit entweder den Zustand $|0\rangle$ oder $|1\rangle$.
- Wenn Alice die 1 übermitteln möchte, dann wählt sie σ_x als Observable. Danach hat Bobs Qubit entweder den Zustand $|+\rangle$ oder $|-\rangle$.

Bob erhält dennoch keine Informationen

Sie werden nun eine heuristische Beweisskizze dafür sehen, wieso durch dieses Protokoll dennoch keine Information übertragen wird, auch wenn Bobs Qubit sich jeweils in den beiden Fällen in einer anderen Menge befindet. Betrachten wir hierzu diese beiden Fälle. Die Schlussfolgerung finden Sie im Anschluss.

Fall 1: Alice möchte die 0 übertragen

Angenommen, dass Alice die 0 übermitteln möchte, d.h., Alice führt eine Messung bezüglich σ_z durch. Bobs Qubit hat also entweder den Zustand $|0\rangle$ oder $|1\rangle$ angenommen. Wie kann Bob nun erfahren, dass Alice ihm eine 0 mitteilen möchte? Natürlich müsste er hierfür irgendetwas messen.

- Angenommen Bob führt eine Messung bezüglich σ_z durch. Dann verändert sich der Zustand seines Qubits nicht, da sowohl $|0\rangle$ als auch $|1\rangle$ Eigenvektoren sind von σ_z. Demnach hat Bob denselben Zustand wie Alice (aufgrund von Gleichung (5.4.1)). Alice hat wiederum mit Wahrscheinlichkeit 1/2 jeweils $|0\rangle$ oder $|1\rangle$ erhalten. Somit hat hier auch Bobs Qubit mit Wahrscheinlichkeit 1/2 jeweils den Zustand $|0\rangle$ oder $|1\rangle$ angenommen.

- Angenommen, Bob führt eine Messung bezüglich σ_x durch. Aufgrund des Born'schen Gesetzes (Lemma 4.5.8) sowie der Identitäten

$$|0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) \quad \text{und} \quad |1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle) \tag{5.4.3}$$

  gilt, dass Bobs Qubit in beiden Fällen – d.h., unabhängig davon, ob das Qubit von Alice nach der Messung den Zustand $|0\rangle$ oder $|1\rangle$ angenommen hat – mit Wahrscheinlichkeit 1/2 jeweils den Zustand $|+\rangle$ oder $|-\rangle$ angenommen hat.

Fall 2: Alice möchte die 1 übertragen

In diesem Fall führt Alice eine Messung bezüglich σ_x durch. Bobs Qubit hat also entweder den Zustand $|+\rangle$ oder $|-\rangle$ angenommen.

- Angenommen, Bob führt eine Messung bezüglich σ_z durch. Aufgrund des Born'schen Gesetzes (Lemma 4.5.8) sowie der Identitäten

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{und} \quad |1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

  gilt, dass Bobs Qubit in beiden Fällen – d.h., unabhängig davon, ob das Qubit von Alice nach der Messung den Zustand $|+\rangle$ oder $|-\rangle$ angenommen hat – mit Wahrscheinlichkeit 1/2 jeweils den Zustand $|0\rangle$ oder $|1\rangle$ angenommen hat.
- Angenommen, Bob führt eine Messung bezüglich σ_x durch. Dann verändert sich der Zustand seines Qubits nicht, da sowohl $|+\rangle$ als auch $|-\rangle$ Eigenvektoren sind von σ_x. Demnach hat Bobs Qubit denselben Zustand wie das von Alice (aufgrund von Gleichung (5.4.2)). Alice hat wiederum mit Wahrscheinlichkeit 1/2 jeweils $|+\rangle$ oder $|-\rangle$ erhalten. Somit hat hier auch Bobs Qubit mit Wahrscheinlichkeit 1/2 jeweils den Zustand $|+\rangle$ oder $|-\rangle$ angenommen.

Schlussfolgerung

Wir sehen also, dass sich in beiden Fällen Bobs Messungen exakt gleich verhalten:

- Misst er bezüglich σ_x, so erhält er **sowohl in Fall 1 als auch in Fall 2** mit Wahrscheinlichkeit 1/2 jeweils den Zustand $|+\rangle$ oder $|-\rangle$.
- Misst er bezüglich σ_z, so erhält er **sowohl in Fall 1 als auch in Fall 2** mit Wahrscheinlichkeit 1/2 jeweils den Zustand $|0\rangle$ oder $|1\rangle$.

Es gibt zwar jeweils verschiedene Gründe, aber das Endresultat – also das finale statistische Verhalten seines Qubits – ist aus Sicht von Bob *in beiden Fällen*

exakt gleich. Um es in einer Metapher zu sagen: Viele verschiedene Wege führen nach Rom, und wenn man mit verbundenen Augen in Rom ankommt und erst dort die Augenbinde abnimmt, weiß man nicht, auf welchem Wege man dorthin gelangt ist.

Daher kann Bob (zumindest mit den bisher genutzten Methoden) nicht detektieren, ob Alice σ_z oder σ_x gewählt hat. Dies liefert eine heuristische Erklärung dafür, wieso dieses Protokoll keine Informationen übermittelt.

Rigoroser Beweis und Verallgemeinerung

Dies ist natürlich kein rigoroser Beweis. So einen Beweis führt man am besten mithilfe sogenannter *gemischter Zustände*, die wir in diesem Buch aber nicht betrachten. Die vorangehenden Ideen liefern aber bereits wichtige Argumente, die im Beweis enthalten sind. Außerdem haben wir das Resultat auch nicht in einer rigorosen Form formuliert. Für eine konkrete Formulierung des Resultats sowie für einen rigorosen Beweis verweise ich Sie auf Abschnitt 4.6.1 in [Sch19].

Ich möchte nochmals betonen, dass wir hier einen konkreten verschränkten Zustand und nur ein mögliches Kommunikationsprotokoll betrachtet haben. Es ist eine interessante Übung, sich zu überlegen, wieso auch bei anderen konkreten Kommunikationsprotokollen sowie bei anderen konkreten Zuständen keine Information übertragen wird.

5.5 Das No-Cloning-Theorem: Kopieren verboten!

Während ich dieses Buch hier schreibe, versuche ich stets Sicherheitskopien zu machen. Ich hoffe, dass auch Sie das bei Ihren wichtigen Dokumenten tun, denn – sei es durch Verlust oder Systemschaden – wir möchten nicht, dass die investierte Arbeit verloren geht. Daher ist es eine legitime Frage, ob dies auch in der Quanteninformatik möglich ist. Die kurze und vielleicht überraschende Antwort lautet aber: Nein, das ist nicht möglich, und zwar aufgrund des sogenannten *No-Cloning-Theorems*, das ich in diesem Abschnitt herleiten werde.

Metaphorische Schilderung der Herausforderung

Es erscheint doch überraschend, dass Kopien in der Quanteninformatik nicht möglich sind. Daher möchte ich mittels einer Metapher das Problem etwas näher erläutern.

Wenn Sie eine Kopie erstellen von einem wichtigen Papierdokument möchten, dann müssen wir es erst scannen und dann drucken. Das neue, ausgedruckte Dokument gleicht dem Originaldokument schon ziemlich gut – je nachdem, wie qualitativ hochwertig unser Kopiergerät ist. Doch ist es wirklich exakt gleich? Was passiert, wenn wir immer tiefer heranzoomen? Werden

die beiden Dokumente sogar bis in die atomare Ebene identisch sein? Es ist klar, dass man einen Hersteller, der so ein präzises Kopiergerät verspricht, zumindest kritisch hinterfragen muss. Dass sich selbst die Atome in ihren jeweiligen Zuständen als Quantenobjekt gleichen können, erscheint mit unseren bisherigen Einblicken in die Quantenmechanik nicht vereinbar zu sein.

Und da sehen Sie auch schon ein Problem beim Kopieren – oder besser gesagt *Klonen* – von Quantenobjekten. Sie werden sehen, dass das Klonen tatsächlich mit den Gesetzen der Quantenmechanik nicht vereinbar ist.

Definition eines Quanten-Kopiergeräts

Vorher muss ich aber erst definieren, was genau wir unter einem *Quanten-Kopiergerät* verstehen. Ich meine damit ein Gerät, das als Input ein beliebiges zu kopierendes Quantensystem mit dem Zustand $|\phi\rangle \in \mathscr{H}$ (aus einem Hilbertraum $\mathscr{H}$) und ein bestimmtes Quantensystem mit dem Zustand $|\psi\rangle \in \mathscr{H}$ erhält und als Output den zusammengesetzten Zustand beider Quantensysteme ausgibt, in dem $|\psi\rangle$ von $|\phi\rangle$ überschrieben wird. Das heißt, hier übernimmt $|\psi\rangle$ die Rolle der leeren Seite. Natürlich muss dieses Gerät eine unitäre Abbildung sein, denn sonst wäre Postulat II.2 verletzt. Wir erhalten hiermit die folgende Definition.

Definition 5.5.1 Sei $\mathscr{H}$ ein Hilbertraum. Ein *Quanten-Kopiergerät* ist eine unitäre Abbildung

$$K : \mathscr{H} \otimes \mathscr{H} \to \mathscr{H} \otimes \mathscr{H},$$

bei der es ein normiertes $|\psi\rangle \in \mathscr{H}$ gibt, sodass

$$K(|\phi\rangle \otimes |\psi\rangle) = |\phi\rangle \otimes |\phi\rangle$$

für alle beliebigen, normierten $|\phi\rangle \in \mathscr{H}$.

Es ist hier wichtig zu betonen, dass von einem Quanten-Kopiergerät verlangt wird, dass es alle Zustände kopieren kann und nicht nur einen Teil aller möglichen Zustände. Beispielsweise kann das sogenannte *CNOT-Gatter* (definiert in Gleichung (7.4.2)) die Zustände $|0\rangle$ und $|1\rangle$ kopieren (wenn man $|\psi\rangle = |0\rangle$ in Definition 5.5.1 wählt).

Das No-Cloning-Theorem

Wir sind nun bereit für das No-Cloning-Theorem. Der Beweis folgt gemäß dem Prinzip *Beweis per Widerspruch*. Das heißt, wir werden zunächst annehmen, dass das Gegenteil von dem, was wir beweisen möchten, gilt. Dann werden wir zeigen, dass unter dieser Annahme Widersprüche impliziert werden.

Proposition 5.5.2 Es gibt kein Quanten-Kopiergerät.

Beweis: Angenommen, ein Quanten-Kopiergerät K existiert.
Seien $|\phi_0\rangle, |\phi_1\rangle, |\psi\rangle \in \mathscr{H}$ mit $\|\phi_0\| = \|\phi_1\| = \|\psi\| = 1$. Dann gilt gemäß Definition 5.5.1, dass

$$K(|\phi_0\rangle \otimes |\psi\rangle) = |\phi_0\rangle \otimes |\phi_0\rangle \quad \text{und} \quad K(|\phi_1\rangle \otimes |\psi\rangle) = |\phi_1\rangle \otimes |\phi_1\rangle .$$

Daraus folgern wir, da K unitär ist und gemäß der Definition des Skalarprodukts beim Tensorprodukt, dass

$$\begin{aligned}\langle\phi_0|\phi_1\rangle &= \langle\phi_0|\phi_1\rangle \cdot 1 = \langle\phi_0|\phi_1\rangle \cdot \langle\psi|\psi\rangle = \langle\, \phi_0 \otimes \psi \mid \phi_1 \otimes \psi \,\rangle \\ &= \langle\, K(|\phi_0\rangle \otimes |\psi\rangle) \mid K(|\phi_1\rangle \otimes |\psi\rangle) \,\rangle \\ &= \langle\, \phi_0 \otimes \phi_0 \mid \phi_1 \otimes \phi_1 \,\rangle \\ &= |\langle\phi_0|\phi_1\rangle|^2.\end{aligned}$$

Es gilt also, dass $\langle\phi_0|\phi_1\rangle = |\langle\phi_0|\phi_1\rangle|^2$. Diese Gleichung gilt aber nur genau dann, wenn $\langle\phi_0|\phi_1\rangle = 0$ oder $\langle\phi_0|\phi_1\rangle = 1$. Dies widerspricht der Bedingung in Definition 5.5.1, dass $|\phi_0\rangle, |\phi_1\rangle$ beliebig gewählt sein können. Somit kann die Annahme, dass ein Quanten-Kopiergerät existiert, nicht gehalten werden. □

5.6 Dichte Quantenkodierung

Sie haben in den letzten beiden Abschnitten gesehen, was mit einem Quantencomputer *nicht* möglich ist. Um die Motivation hochzuhalten, wollen wir in diesem Abschnitt ein erstes Gefühl dafür bekommen, dass tatsächlich erstaunliche Dinge möglich sein können, wenn man die quantenmechanischen Phänomene geschickt auszunutzen weiß. Daher möchte ich ein Kommunikationsprotokoll zwischen Alice und Bob vorstellen, bei dem Alice Bob *ein* Qubit schickt, aber Bob die Information von *zwei* (klassischen) Bits erhält. Derartige Kommunikationsprotokolle (bei denen weniger Qubits geschickt werden als Bits an Informationen) nennt man *dichte Quantenkodierung*.

Ich werde im Folgenden erst die Voraussetzungen für dieses Kommunikationsprotokoll auflisten, dann das Kommunikationsprotokoll beschreiben und dessen Korrektheit nachweisen. Zum Schluss ordne ich dieses Resultat noch praktisch ein.

Bemerkung 5.6.1 Eine mögliche Quelle für Verwirrung (auf die ich Sie hiermit hinweise) kann im Folgenden die Tatsache sein, dass wir die Pauli-Matrizen sowohl als unitäre Matrizen zur Manipulation von Qubits als auch als Observable nutzen werden.

Voraussetzungen

Genauso wie in Abschnitt 5.4 nehmen wir in diesem Abschnitt an, dass Alice und Bob ein verschränktes Zwei-Qubit-Register im Zustand $|\Phi_+\rangle$ teilen, wobei jeder von ihnen Zugang zu einem Qubit hat. Wir notieren außerdem mit $b_1 b_0$ die zwei Bits, die Alice an Bob übermitteln möchte. Es gilt, dass $b_1 b_0 \in \{00, 01, 10, 11\}$. Letztlich nehmen wir auch an, dass Alice und Bob sich vorab auf das folgende Kommunikationsprotokoll und dessen Interpretation geeinigt haben.

Kommunikationsprotokoll

Die Idee des Kommunikationsprotokolls lautet, dass Alice ihr Qubit durch eine unitäre Matrix manipuliert, wobei diese unitäre Matrix davon abhängt, welche Bits sie an Bob schicken möchte. Anschließend schickt sie ihr Qubit an Bob, und dieser führt geeignete Messungen durch, durch die er rekonstruieren kann, welche unitäre Matrix Alice gewählt hat. Da diese Wahl von den Bits abhängt, die Alice schicken möchte, findet er so die gesuchten Bitwerte heraus. Ein wichtiger Punkt für die Korrektheit dieses Kommunikationsprotokolls (die wir später nachprüfen werden) lautet, dass die unitären Matrizen, die Alice wählt, jeweils immer andere Messergebnisse liefern für Bob.

Im Detail lautet das Kommunikationsprotokoll wie folgt:

- Alice wendet eine unitäre Abbildung $U_{b_1 b_0} \in \mathcal{U}(\mathcal{H})$ auf das ihr vorliegende Qubit an, wobei $U_{b_1 b_0}$ definiert ist durch

  $$U_{b_1 b_0} = \sigma_z^{b_0} \cdot \sigma_x^{b_1}.$$

  In der Definition von $U_{b_1 b_0}$ nutzen wir die Konvention, dass für alle Matrizen A gilt, dass $A^0 = \mathrm{Id}$.
- Alice sendet ihr Qubit an Bob.
- Bob führt zwei Messungen durch: erst bezüglich $\sigma_z \otimes \sigma_z$ und dann bezüglich $\sigma_x \otimes \sigma_x$.
- Aus den Messergebnissen extrahiert Bob die von Alice verschickten Bitwerte gemäß des Schemas in Tabelle 5.1.

Bobs Messergebnisse	Zugeordnete Bitwerte $b_1 b_0$
$+1, +1$	00
$+1, -1$	01
$-1, +1$	10
$-1, -1$	11

Tabelle 5.1 Bobs Auswahlschema

Korrektheit des Kommunikationsprotokolls

Um die Korrektheit dieses Protokolls zu prüfen, müssen wir im Wesentlichen nur die Plausibilität der Tabelle 5.1 verifizieren. Hierzu müssen wir sichergehen, dass die besagten Ergebnisse tatsächlich mit den jeweiligen Bitwerten korrespondieren, die Alice schicken möchte.

Um die Messergebnisse nachvollziehen zu können, müssen wir zunächst den Zustand des Zwei-Qubit-Registers vor der Messung berechnen. Dieser ist gegeben durch

$$\begin{aligned}(\sigma_z^{b_1}\sigma_x^{b_0}\otimes \mathrm{Id})\,|\Phi_+\rangle &= (\sigma_z^{b_1}\sigma_x^{b_0}\otimes \mathrm{Id})\left(\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)\right)\\ &= (\sigma_z^{b_1}\sigma_x^{b_0}\otimes \mathrm{Id})\left(\frac{1}{\sqrt{2}}(|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle)\right)\\ &= \frac{1}{\sqrt{2}}\Big((\sigma_z^{b_1}\sigma_x^{b_0}\otimes \mathrm{Id})\,|0\rangle\otimes|0\rangle+(\sigma_z^{b_1}\sigma_x^{b_0}\otimes \mathrm{Id})\,|1\rangle\otimes|1\rangle)\Big)\\ &= \frac{1}{\sqrt{2}}\Big((\sigma_z^{b_1}\sigma_x^{b_0}\,|0\rangle)\otimes|0\rangle+(\sigma_z^{b_1}\sigma_x^{b_0}\,|1\rangle)\otimes|1\rangle)\Big).\end{aligned}$$

Im Fall $b_1 = b_0 = 1$ ergibt sich aufgrund der Identitäten in Lemma 4.4.9 und gemäß der Definition der Bell-Basis in Gleichung (4.6.18), dass

$$\begin{aligned}(\sigma_z^{b_1}\sigma_x^{b_0}\otimes \mathrm{Id})\,|\Phi_+\rangle &= \frac{1}{\sqrt{2}}\Big((\sigma_z\sigma_x\,|0\rangle)\otimes|0\rangle+(\sigma_z\sigma_x\,|1\rangle)\otimes|1\rangle)\Big)\\ &= \frac{1}{\sqrt{2}}\Big((\sigma_z\,|1\rangle)\otimes|0\rangle+(\sigma_z\,|0\rangle)\otimes|1\rangle)\Big)\\ &= \frac{1}{\sqrt{2}}\Big(-|1\rangle\otimes|0\rangle+|0\rangle\otimes|1\rangle)\Big)\\ &= \frac{1}{\sqrt{2}}\Big(-|10\rangle+|01\rangle)\Big) = |\Psi_-\rangle\,.\end{aligned}$$

Für die restlichen Kombinationsmöglichkeiten der Bitwerte $b_1 b_0$ ergeben sich die Zustände in Tabelle 5.2. Den Beweis der noch nicht durchgerechneten Behauptungen finden Sie in Übung 5.8.

$b_1 b_0$	$(\sigma_z^{b_0}\sigma_x^{b_1}\otimes \mathrm{Id})\,\|\Phi_+\rangle$
00	$\|\Phi_+\rangle$
01	$\|\Psi_+\rangle$
10	$\|\Phi_-\rangle$
11	$\|\Psi_-\rangle$

Tabelle 5.2 Berechnung der Zustände $(\sigma_z^{b_0}\sigma_x^{b_1}\otimes \mathrm{Id})\,|\Phi_+\rangle$

Als Nächstes müssen wir prüfen, welche Eigenvektoren und Eigenwerte die beiden Observablen $\sigma_z\otimes\sigma_z$ und $\sigma_x\otimes\sigma_x$ haben, da Messungen durch die

Eigenräume charakterisiert sind. Hierzu gelten die Zusammenhänge in Tabelle 5.3, die wir in Übung 5.9 (wieder mithilfe der Identitäten in Lemma 4.4.9) nachprüfen:

Bell-Basis-Element	Eigenwert als Eigenvektor von $\sigma_z \otimes \sigma_z$	Eigenwert als Eigenvektor von $\sigma_x \otimes \sigma_x$
$\|\Phi_+\rangle$	$+1$	$+1$
$\|\Phi_-\rangle$	$+1$	-1
$\|\Psi_+\rangle$	-1	$+1$
$\|\Psi_-\rangle$	-1	-1

Tabelle 5.3 Zuordnung der Messergebnisse von Bob zu den Bell-Basis-Elementen

Somit gilt, dass Bob aufgrund seiner Messergebnisse eindeutig herausfinden kann (mittels der Tabelle 5.3), welchen Zustand das Zwei-Qubit-Register vor der Messung innehatte. Dadurch weiß er aber wiederum aufgrund von Tabelle 5.2, welche unitäre Matrix $\sigma_z^{b_0}\sigma_x^{b_1}$ Alice angewendet haben muss. Und aus dieser Auswahl der unitären Matrix kann er direkt die Bits extrahieren, die Alice ihm übermitteln wollte.

Hierdurch erhalten wir die Plausibilität der Tabelle 5.1, die einfach eine Kurzfassung der Tabellen 5.2 und 5.3 ist.

Praktische Einordnung

In der Praxis muss man bei der Ausnutzung dieses Kommunikationsprotokolls berücksichtigen,

- dass bereits ein verschränkter Zustand zwischen Alice und Bob existieren muss und
- dass ein Qubit fehlerfrei von Alice nach Bob transferiert werden muss.

Diese beiden Voraussetzungen sind in der heutigen Infrastruktur natürlich aufwendiger, als zwei Bits zu übertragen. Aber man weiß nie, wie die Welt in der Zukunft aussehen wird.

5.7 Quantenteleportation

Wohl jeder Mensch hat sich wohl schon einmal gedacht: »Ich wünschte, ich könnte mich jetzt nach Hause beamen lassen.« In diesem Abschnitt sehen Sie, dass dies – zu einem sehr sehr hohen Preis – in der Theorie vielleicht gar nicht

- Die meiste Kreativität wird uns abverlangt, wenn der Fall *f ist balanciert* vorliegt – im Code gekennzeichnet durch den Parameter `balanciert`. Zum Verständnis, warum die wie im Code angegebene Anwendung des CNOT-Gatters zum Erfolg führt, ist das Konzept des *Hamming-Gewichts* ω_H eines n-Bit-Strings k hilfreich. Das Hamming-Gewicht von k ist definiert als die Anzahl der Bits in k, die den Wert 1 haben, d. h.,

$$\omega_H(k) = |\{k_i \mid k_i = 1\}|.$$

  Wo findet sich das Hamming-Gewicht im Code wieder? Das sehen Sie jetzt.

  Vor Schritt (3) befindet sich das n-Qubit-Register in der gleichförmigen Superposition. Aufgrund von Linearität genügt es, uns anzuschauen, was im Fall `balanciert` mit einem einzelnen Summanden dieser Superposition passiert. Sei $|k\rangle$ ein Element der Rechenbasis und somit ein Summand in dieser Superpostion. Wir betrachten also den Zustand

$$|k\rangle = |k_{n-1}\rangle \otimes \cdots \otimes |k_0\rangle$$

  Im Code wird nun nacheinander jedes einzelne dieser $|k_0\rangle, \ldots, |k_{n-1}\rangle$ als Kontroll-Qubit verwendet. Wann immer eines dieser Qubits gleich dem Zustand $|1\rangle$ ist – was genau $\omega_H(k)$-mal der Fall ist (!) –, wird die Pauli-X-Matrix auf das Hilfsregister angewendet und somit eine globale Phase (-1) induziert. Daher erhalten wir nach der Schleife im Code

$$(-1)^{\omega_H(k)} |k\rangle .$$

  Und aufgrund von Linearität erhalten wir für den gesamten Zustand nach dieser Schleife

$$\sum_{k=0}^{2^n-1} (-1)^{\omega_H(k)} |k\rangle .$$

  Es bleibt nur noch zu zeigen, dass $(-1)^{\omega_H(k)}$ für genau die Hälfte aller k den Wert 1 ergibt und für die andere Hälfte den Wert (-1). Um diese Aufgabe kümmern wir uns in einer Übung.

9.5 Der Bernstein-Vazirani-Algorithmus

Ich habe bereits erwähnt, dass der Deutsch-Jozsa-Algorithmus die Ungleichheit der Problemklassen **P** und **EQP** aufzeigt. Dies wird als *Separation* der beiden Klassen bezeichnet. Es stellt sich noch die Frage, wie das Verhältnis ist von

- **BQP** (*Bounded-Error Quantum Polynomial Time*), der Klasse von Problemen, die effizient und *mit einer Fehlerwahrscheinlichkeit kleiner oder gleich 1/3* von einem Quantencomputer gelöst werden können,

zu

- **BPP** (*Bounded-Error Probabilistic Polynomial Time*), der Klasse von Problemen, die effizient, *probabilistisch* und mit einer Fehlerwahrscheinlichkeit kleiner oder gleich 1/3 von einem klassischen Computer gelöst werden können.

Es sei erwähnt, dass der Wert 1/3 hier nur Konvention ist und man auch andere Werte hätte wählen können (siehe [AB09] für mehr Details).

Dieselben Argumente, die **P** $\subset$ **EQP** gezeigt haben, implizieren auch **BPP** $\subset$ **BQP**. Diesmal lässt sich aber mittels des Deutsch-Jozsa-Algorithmus darüber hinaus zu diesem Verhältnis keine Aussage treffen. Denn auch klassische Computer könnten das Deutsch-Jozsa-Problem lösen, wenn die Anforderung *mit Wahrscheinlichkeit 1* abgeschwächt würde und die Korrektheit des Algorithmus nur *mit hoher Wahrscheinlichkeit* verlangt wäre.

Wenig später, in der Arbeit [BV93] von 1993, die 1997 überarbeitet erschien, zeigten Bernstein und Vazirani dahingehend interessante Resultate. Sie demonstrierten, dass eine eingeschränkte Version des Deutsch-Jozsa-Algorithmus ein anderes Problem, das sogenannte *Bernstein-Vazirani-Problem*, löst, während auf einem klassischen Computer deutlich mehr Schritte nötig wären, um dieses Problem zu lösen.

Sie zeigten des Weiteren eine *Orakel-Separierung* zwischen **BPP** und **BQP**. Das heißt, dass – unter der Annahme, dass auf ein Orakel zugegriffen wird – es in **BQP** ein Problem gibt, das unter der gleichen Annahme nicht in **BPP** liegt.

In diesem Abschnitt beschäftigen wir uns mit dem Bernstein-Vazirani-Problem und mit dessen klassischer sowie quantenbasierter Lösung.

Definition 9.5.1 Gegeben sei eine Boole'sche Funktion $f : \{0,1\}^n \to \{0,1\}$, die mittels eines geheimen Schlüssels $s \in \{0,1\}^n$ definiert ist durch

$$f(k) = k \odot s$$

für alle $k \in \{0,1\}^n$. Das *Bernstein-Vazirani-Problem* besteht darin, mit möglichst wenigen Evaluierungen der Funktion f oder eines zugehörigen Orakels und mit einer Fehlerwahrscheinlichkeit kleiner oder gleich 1/3 den Schlüssel s zu extrahieren.

Das Bernstein-Vazirani-Problem mittels klassischer Computer

Ein ziemlich einfacher, klassischer Algorithmus besteht darin, alle Strings der Form

$$00\cdots01,\ 0\cdots010,\quad \ldots\quad ,010\cdots0,\ 10\cdots0$$

in f einzusetzen, da für $j \in \{0,\ldots,n-1\}$,

$$f(0\cdots010\cdots0) = s_j$$

wobei die 1 im Eingabewert von f an j-ter Stelle steht und s_j das j-te Bit von s ist. Dieser Ansatz würde uns n Evaluierungen von f kosten. Wir möchten aber die Anzahl der Evaluierungen von f minimieren, daher sollten wir uns noch ein paar Gedanken darüber machen, ob wir unseren Algorithmus optimieren könnten. Möglicherweise auch zu dem Preis, dass wir nur noch mit einer ausreichend großen Wahrscheinlichkeit das richtige s erhalten.

Bis jetzt ist kein klassischer Algorithmus bekannt, der mit weniger als $O(n)$ Evaluierungen von f das Bernstein-Vazirani-Problem löst. Intuitiv lässt es sich damit begründen, dass man mittels $O(n)$ klassischer Aufrufe auch nur höchstens $O(n)$ Bits an Informationen gewinnen kann.

Der Bernstein-Vazirani-Algorithmus

Der *Bernstein-Vazirani-Algorithmus* zur Lösung des Bernstein-Vazirani-Problems ist ein Quantenalgorithmus, der – unter der Annahme, dass Zugriff auf das Orakel $\mathcal{O}_f$ zu f existiert – aus exakt denselben Schritten (1), (2), (3) und (4) des Deutsch-Jozsa-Algorithmus besteht. Erfreulicherweise ist im Anschluss an diese Schritte bereits das n-Qubit-Register im Zustand $|s\rangle$! Wieso das so ist, sehen Sie in Kürze. Vorher möchten ich nur festhalten, dass wir auch hier nur einen Aufruf der Orakelfunktion zu f benötigt haben.

Wieso löst aber der Bernstein-Vazirani-Algorithmus das Bernstein-Vazirani-Problem? In anderen Worten: Wieso ist nach Schritt (4) das n-Qubit-Register im Zustand $|s\rangle$? Das sehen wir daran, dass wir gemäß der Berechnungen oben wissen, dass nach Schritt (3) das n-Qubit-Register sich in folgendem Zustand befindet:

$$\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\,|j\rangle = \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{j\odot s}\,|j\rangle = H^{\otimes n}(|s\rangle),$$

wobei wir im letzten Schritt Lemma 9.2.2 benutzt haben. Aber auch eben dieses Lemma sagt uns, dass wir nach der abermaligen Anwendung von $H^{\otimes n}$ in Schritt (4) den Zustand $|s\rangle$ erhalten, da $H^{\otimes n}$ selbstinvers ist. Dies liefert uns die Korrektheit des Algorithmus.

Sie sehen also, dass das Bernstein-Vazirani-Algorithmus zeigt, dass Quantencomputer das Bernstein-Vazirani-Problem polynomiell schneller lösen können als klassische Computer. Unsere nächste Begegnung mit der Quantenüberlegenheit!

9.6 Übungen und bibliografische Hinweise

9.6.1 Übungen

Übung 9.1 Zeigen Sie, dass $H^{\otimes n}$ selbstinvers ist.

Übung 9.2 Implementieren Sie den Bernstein-Vazirani-Algorithmus.

9.6.2 Bibliografische Hinweise

Die Referenzen zu den beiden in diesem Kapitel vorgestellten Quantenalgorithmen lauten: [DJ92] und [BV93]. Alternativ sind diese Algorithmen natürlich auch in den gängigen Textbüchern zur Quanteninformatik beschrieben. Ich empfehle Ihnen insbesondere die Beschreibung des Deutsch-Jozsa-Algorithmus aus [Sch19].

Index

J

K

L

M

N

O

P

Q

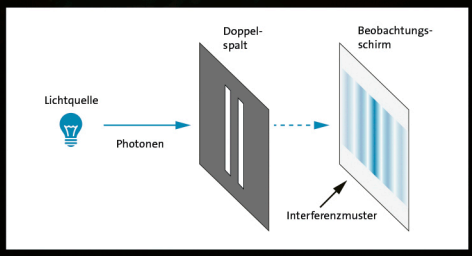
QUANTENCOMPUTING

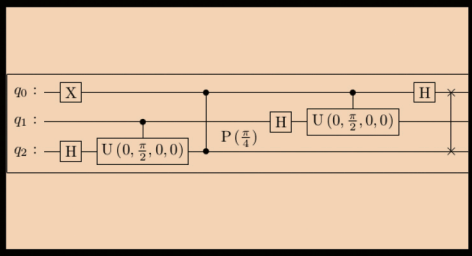
Qubits können mehr als klassische Bits: Sie können mehrere Zustände überlagert annehmen und sich auf »spukhafte« Weise miteinander verbinden. Diese quantenmechanischen Phänomene und ihre mathematischen Hintergründe erklärt Ihnen Dr. Kaveh Bashiri. Sie lernen, wieso es kein Quanten-Kopiergerät geben kann, wie Teleportation in der Welt der Quanten gelingt und was das alles mit der Sicherheit kryptografischer Algorithmen zu tun hat.

Auf diesem Fundament starten Sie eigene Programmierexperimente, ohne dass Sie die Ausstattung eines Forschungslabors brauchen. Mit Qiskit emulieren Sie die Funktion echter Quantenrechner und probieren Quantenalgorithmen aus. Jupyter Notebooks stehen zum Download bereit.

Aus dem Inhalt

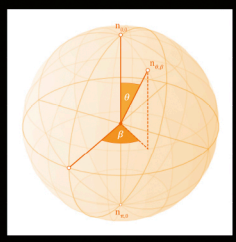
- Was sind Quantencomputer?
- Mathematische Grundlagen
- Von der klassischen Informatik zur Quanteninformatik
- Elemente der Quantenmechanik: Zustände, Messungen und Tensorprodukte
- Unschärferelation und Quantenverschränkungen
- No-Communication-, No-Cloning-Theorem
- Quantenschaltkreismodelle und Qubits
- Dichte Quantenkodierung
- Quantenteleportation
- Algorithmen und Anwendung: Deutsch-Jozsa, Simon, Shor, Grover
- Post-Quanten-Kryptografie





$$|m\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{\ell=0}^{2^n-1} e^{2\pi i \ell m \frac{1}{2^n}} |\ell\rangle$$

Dr. Kaveh Bashiri hat in der Mathematischen Physik promoviert. Seitdem beschäftigt er sich mit den kryptografischen Auswirkungen von Quantenalgorithmen und mit Post-Quanten-Verfahren. In diesem Buch begleitet er Sie mit vielen Codebeispielen und Übungsaufgaben in die Welt der Quanten.



Rheinwerk Computing

Gedruckt in Deutschland
Mineralölfreie Druckfarben
Zertifiziertes Papier

€ 39,90 [D] € 41,10 [A]

Informatik
ISBN 978-3-8362-9848-3