

UNIVERSITÄT DER BUNDESWEHR MÜNCHEN
Fakultät für Elektrotechnik und Informationstechnik

Stochastische Analyse von Quantenalgorithmen

Robert Schmied

Vorsitzender des Promotionsausschusses: Prof. Dr.-Ing. Landes

1. Berichterstatter: Prof. Dr. rer. nat. Dr.-Ing. Schäffler

2. Berichterstatter: Priv.-Doz. Dr.-Ing. Schulze

Tag der Prüfung: 23. Mai 2005

Mit der Promotion erlangter akademischer Grad:
Doktor-Ingenieur
(Dr.-Ing.)

Neubiberg, den 23. Mai 2005

UNIVERSITÄT DER BUNDESWEHR MÜNCHEN
Fakultät für Elektrotechnik und Informationstechnik

Stochastische Analyse von Quantenalgorithmen

Robert Schmied

Vorsitzender des Promotionsausschusses: Prof. Dr.-Ing. Landes

1. Berichterstatter: Prof. Dr. rer. nat. Dr.-Ing. Schäffler

2. Berichterstatter: Priv.-Doz. Dr.-Ing. Schulze

Tag der Prüfung: 23. Mai 2005

Mit der Promotion erlangter akademischer Grad:

Doktor-Ingenieur

(Dr.-Ing.)

Neubiberg, den 23. Mai 2005

meinen Eltern Christina und Bruno

Inhaltsverzeichnis

Notationen, wichtige Definitionen und Abkürzungen	1
Notationen	1
Definitionen	4
Abkürzungen	5
Motivation	6
Einleitung	6
Aufbau der Arbeit	7
1 Quantenmechanische Grundlagen	10
1.1 Physikalischer Zustandsraum	10
1.2 Operatoren	14
1.3 Zeitliche Dynamik des Systems	16
1.4 Zusammenfassung	18
2 Quantum Computation	20
2.1 Mathematische Modellierung	20
2.1.1 Mathematischer Zustandsraum	20
2.1.2 Modellierung der zeitlichen Dynamik	25
2.2 Konstruktion von Gates	26
2.2.1 Operationen auf einzelnen Bits	26
2.2.2 Bit-m-Operationen	28
2.3 Zerlegung von Gates	31
2.3.1 Zerlegbarkeit von Bit-m-Operationen	32
2.3.2 Zerlegungsalgorithmus	38
2.4 Zusammenfassung	40
3 Quantenalgorithmen	42
3.1 Charakteristika von Quantenalgorithmen	42
3.1.1 Zeitdiskretes Schaltkreis-Modell	43
3.1.2 Adiabatische Quantenalgorithmen	45
3.1.3 Betrachtung als stochastischer Algorithmus	46
3.2 Diskretisierung von Quantenalgorithmen	48

3.3	Klassifikation von Quantenalgorithmen	50
3.3.1	Hidden Subgroup-Probleme	50
3.3.2	Amplitudenverstärkung	52
3.3.3	Quantum-(Random)-Walk	53
3.4	Zusammenfassung	56
4	Simulation mit einem Branch&Bound-Verfahren	57
4.1	Algorithmusidee	57
4.2	Umsetzung des Verfahrens	59
4.3	Güte des Verfahrens	60
4.3.1	Problemdefinition	60
4.3.2	Abschätzung der Änderungsmöglichkeiten	63
4.3.3	„best case“-Abschätzung	64
4.3.4	„worst case“-Abschätzung	65
4.3.5	„average case“-Abschätzung	67
4.4	Zusammenfassung	69
5	Simulationsergebnisse	71
5.1	Suchalgorithmus von Grover	72
5.2	Quantum-Walk	75
6	Zusammenfassung und Ausblick	85
Anhang		87
	Abbildungsverzeichnis	87
	Tabellenverzeichnis	88
	Literaturverzeichnis	90
	Index	92

Notationen, wichtige Definitionen und Abkürzungen

Die nachfolgende Auflistung ist eine Sammlung häufig gebrauchter und wichtiger Notationen, Definitionen und verwendeter Abkürzungen.

Dirac-Notation

Die von P.A.M. Dirac¹ eingeführte Schreibweise für Vektoren und Operatoren in der Quantenmechanik wird sehr gerne angewendet und soll hier kurz erwähnt werden. Da sie jedoch in der Literatur oftmals nicht eindeutig gebraucht wird, soll die Notation möglichst vermieden werden. In einigen Fällen kann es für die Schreibung einfacher sein wie etwa zu Beginn des Abschnittes 3.1.1.

komplexe Vektoren	Dirac
ψ	$ \psi\rangle$
ϕ^\dagger	$\langle\phi $
$\phi^\dagger\psi$	$\langle\phi \psi\rangle$
$\psi \leftrightarrow \psi^\dagger$	$ \psi\rangle \leftrightarrow \langle\psi $
$A\psi$	$A \psi\rangle$
$\phi^\dagger A$	$\langle\phi A$
$(A\phi)^\dagger \equiv \phi^\dagger A^\dagger$	$\langle A\phi \equiv \langle\phi A$
$(A\phi)^\dagger\psi \equiv \phi^\dagger A^\dagger\psi$	$\langle A\phi \psi\rangle \equiv \langle\phi A \psi\rangle$
$\psi_1 \otimes \psi_2$	$ \psi_1\psi_2\rangle = \psi_1\rangle \otimes \psi_2\rangle$

Tafel 1: Vektoren- und Dirac-Notation im Vergleich [Pere93].

Notationen

Hier werden die in der Arbeit häufig auftretenden bzw. wichtigen Bezeichnungen und mathematischen Notationen aufgeführt.

¹Paul Adrien Maurice Dirac, 08.08.1902-20.10.1984

Symbol	Bedeutung
def	Definition
$\mathcal{O}(\cdot)$	Landau-Symbol: Komplexitätsordnung
\forall	All-Quantor
\in, \notin	Elementbeziehung bzw. Nichtelementbeziehung
\Rightarrow	Folgerung
\Leftrightarrow	Äquivalenzsymbol
\sim	Äquivalenz einer Äquivalenzrelation
\uparrow, \downarrow	Spin-up- bzw. Spin-down-Teilchen
$=, \neq$	Gleichheit bzw. Ungleichheit
$+, -, \cdot, \pm$	Additions-, Subtraktions-, Multiplikations- und Plusminusussymbol
$\sqrt{\cdot}$	Quadratwurzel
$\mathbb{R}, \mathbb{C}, \mathbb{N}$	Reelle, komplexe und natürliche Zahlen
Re	Realteil einer Zahl
-	Konjugiert-komplexe Zahl, in Matrizen komponentenweise Betrachtung
$<, \leq, >, \geq$	Vergleichsoperatoren
\hbar	$\frac{h}{2\pi}$ mit Planck'schem Wirkungsquantum $h = 6.6260755 * 10^{-34} J_s$
π	Kreiszahl $\pi = 3.14159265\dots$
e	Eulersche Zahl, $e = 2.7182818\dots$
i	Imaginäre Einheit
∂	Differentialoperator
mod	Operator zur Modulo-Rechnung
div	Operator für Division ohne Rest
max, min	Maximum bzw. Minimum
$[\cdot, \cdot]$	Abgeschlossenes Intervall
$\langle \cdot, \cdot \rangle_V$	Skalarprodukt in einem Vektorraum V
$ \cdot $	Betrag einer Zahl
$\ \cdot\ _k$	k -Norm

$\{ \}, \{a_1, \dots, a_k\}$	Mengensymbol, Menge mit k Elementen
\emptyset	leere Menge
$\subset, \subseteq, \supset, \supseteq$	Teilmengensymbole
A^c	Komplement einer Menge
$ A $	Mächtigkeit einer Menge
$\mathcal{P}(\cdot)$	Potenzmenge einer Menge
$\cap, \bigcap_{i=1}^n$	Durchschnitt von Mengen
$\cup, \bigcup_{i=1}^n$	Vereinigung von Mengen
\vec{x}	Physikalischer Vektor
$x, x^{(t)} \in \mathcal{H}^{\otimes n}$	Komplexer Vektor der Dimension 2^n
$x_j^{(t)}$	Vektorkomponente des Vektors $x^{(t)}$, Eintrag j
$\mathcal{S}_{\mathcal{H}^{\otimes n}}$	Sphäre in einem Hilbertraum
\otimes	Tensorprodukt
$H \in \mathcal{H}^{k,k}$	Komplexe Matrix der Dimension $k \times k$
$m_{ij}^{(t)}$	Matrizeintrag der Matrix $M_t = (m_{ij}^{(t)})$, Zeile i , Spalte j
\dagger, T	Adjungierte bzw. Transponierte einer Matrix
$\text{diag}(x_1, \dots, x_k)$	Diagonalmatrix mit x_i -Einträgen auf der Hauptdiagonalen
\det	Determinante einer Matrix
$\mathbb{1}_n$	Einheitsmatrix der Dimension $n \times n$
$[A, B]$	Kommutator zweier Matrizen
$\sum_{i=1}^n$	Summensymbol
$\prod_{i=1}^n$	Produktsymbol
\rightarrow, \mapsto	Abbildungssymbol und -vorschrift
\int	Integrationsymbol
$f^{-1}(A)$	Urbild einer Funktion
$L^2(\mathbb{R}^k)$	Funktionsraum mit endlicher 2-Norm
$\mathcal{L}^2(\mathbb{R}^k)$	Faktorraum: Hilbertraum mit 2-Norm
(Ω, \mathcal{A}, P)	Wahrscheinlichkeitsraum
$a \wedge b$	Wahrscheinlichkeitenaufteilung im binären Verzweigungsprozess
$E[X], V[X]$	Erwartungswert und Varianz einer Zufallsvariablen X

Definitionen

Alle hier aufgeführten Definitionen² sind für alle Kapitel von zentraler Bedeutung. In der mathematischen Formulierung der Quantenmechanik sind Hilbertraum und stochastische Grundbegriffe unabdingbar. Deshalb werden sie hier im voraus aufgeführt.

Definition 1 (Hilbertraum) Ein Vektorraum $(V, +, *, \langle \cdot, \cdot \rangle_V)$ über dem Körper \mathbb{R} oder \mathbb{C} zusammen mit einem Skalarprodukt $\langle \cdot, \cdot \rangle_V$, in dem jede Cauchy-Folge gegen ein Element des Raumes konvergiert, heißt **Hilbertraum**.

Definition 2 (Ergebnisraum, Elementarereignis, Ereignis)

- **Ergebnisraum.** $\Omega = \{\text{Menge der möglichen Ergebnisse}\} \neq \emptyset$
- **Elementarereignis.** $\omega \in \Omega$
- **Ereignis.** $A \subseteq \Omega$. A tritt ein, wenn ein Elementarereignis $\omega \in A$ als Beobachtung eines Zufallsexperimentes eingetreten ist.

Definition 3 (σ -Algebra, Messraum, Wahrscheinlichkeitsmaß)

- **σ -Algebra.** Mengensystem $\mathcal{A} \subseteq \mathcal{P}(\Omega)$ ($\mathcal{P}(\Omega) \stackrel{\text{def}}{=} \text{Potenzmenge von } \Omega$) mit
 - $\Omega \in \mathcal{A}$
 - $\forall A \in \mathcal{A} : A^c \stackrel{\text{def}}{=} \Omega \setminus A \in \mathcal{A}$
 - $\forall A_n \in \mathcal{A}, n \in \mathbb{N} : \bigcup_{n=1}^{\infty} A_n \in \mathcal{A}$
- **Messraum.** (Ω, \mathcal{A})
- **Wahrscheinlichkeitsmaß.** $P : \mathcal{A} \rightarrow [0, 1]$ mit
 - $P(\Omega) = 1$
 - $P(A) \geq 0 \quad \forall A \in \mathcal{A}$
 - $\forall \{A_n\}_{n \in \mathbb{N}}$ paarweise disjunkt : $P\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n)$

²vgl. [Schä02] und [MeiS05]

Definition 4 (Wahrscheinlichkeitsraum, Messbare Abbildung)

- **Wahrscheinlichkeitsraum.** (Ω, \mathcal{A}, P) , mathematische Modellierung eines Zufallsexperimentes.
- **Messbare Abbildung.** $f : \Omega_1 \rightarrow \Omega_2$ mit $(\Omega_1, \mathcal{A}_1), (\Omega_2, \mathcal{A}_2)$ heißt $\mathcal{A}_1 - \mathcal{A}_2$ -messbar, falls $f^{-1}(A_2) \subset \mathcal{A}_1$.

Definition 5 (Zufallsvariable, Verteilung einer Zufallsvariablen)

- **Zufallsvariable.** $\mathcal{A}_1 - \mathcal{A}_2$ -messbare Abbildung $X : \Omega_1 \rightarrow \Omega_2$ mit $(\Omega_1, \mathcal{A}_1, P), (\Omega_2, \mathcal{A}_2)$.
- **Verteilung einer Zufallsvariablen.** $P_X : \mathcal{A}_2 \rightarrow [0, 1]$, $P_X(B) \stackrel{\text{def}}{=} P(X^{-1}(B))$, $B \in \mathcal{A}_2$, wobei $X : \Omega_1 \rightarrow \Omega_2$ Zufallsvariable mit $(\Omega_1, \mathcal{A}_1, P), (\Omega_2, \mathcal{A}_2)$ ist.

Abkürzungen

Die folgenden sich häufig wiederholenden Begriffe sollen im Text abgekürzt werden.

Abkürzung	Bedeutung
QA	Quantenalgorithmus, -men
QM	Quantenmechanik
QC	Quantum Computation
QFT	Quantum Fourier Transformation

Motivation

Einleitung

Anwendung von Quantenalgorithmen

Quantenalgorithmen (QA) sorgen in jüngster Zeit immer öfter für Aufsehen. In der Kryptographie etwa liefert der Faktorisierungsalgorithmus von Shor viel Gesprächsstoff. Ein bekanntes Schlagwort ist der „exponentielle Speedup“, der mit solchen Algorithmen möglich sei. Doch bei aller Euphorie: Eine Nutzung derartiger Möglichkeiten im Alltag ist bei weitem noch nicht in Sicht. Das liegt daran, dass sich die Vorteile, die sich aus der Quantentheorie ergeben, wieder aufheben, sobald eine Umsetzung an heute vorhandener Rechnertechnologie versucht wird.

Der „Universelle Quantencomputer“ (siehe [Chil00]) ist zwar theoretisch möglich, die praktische Umsetzung birgt aber einige Probleme in sich. So kann beispielsweise kein System völlig von äußeren Einflüssen isoliert werden und das führt gerade in diesem mikroskopischen Bereich zu fehlererzeugenden Einflüssen.

Von der Quantenmechanik zum Quantum Computation

Seit der Entdeckung quantenmechanischer Zusammenhänge zu Beginn des 20. Jahrhunderts, die sich zwar beobachten, aber nur schwer in das rationale Bewusstsein einfügen lassen (z.B. Welle-Teilchen-Dualismus), konnten in den letzten Jahren Eigenschaften dieser „kontraintuitiven Wissenschaft“ [Müth99] zur Anwendung gebracht werden. Jedoch erst mit dem fächerübergreifenden Zusammenspiel von Physik, Mathematik und Informatik entwickelte sich der Begriff des „Quantum Computation“.

Spin-Eigenschaften von Teilchen machen es physikalisch möglich, eine quantentheoretische Imitation der binären Computertechnik aus der Informatik einzuführen. Das funktioniert in der Theorie, die praktische Umsetzung eines „Quantenrechners“ aber liegt noch in den Kinderschuhen. Denn es ist trotz der intensiven Forschung heute noch immer nicht möglich, adäquate physikalische Aufbauten herzustellen. Zum jetzigen Stand sind Quantenrechner mit etwa zehn Bits ([Däne02]), also etwas mehr als einem Byte und so viel zu wenig für die moderne Computergesellschaft, realisiert.

Damit ist ein großes Problem aufgedeckt: Die ungeheuren Möglichkeiten, die sich mit den bereits aufgestellten und den sich in der Zukunft ergebenden QA eröffnen könnten, sind nicht nutzbar. Zum einen können sie nicht auf unseren klassischen Rechnern effizient implementiert werden, zum anderen gibt es auf absehbare Zeit keine Quantenrechner in brauchbarer Größe.

Das führt zu der Idee, gewisse Strukturen, die sich in der Konstruktion von Quantenalgorithmen ergeben, mathematisch aufzugreifen und zu untersuchen, ob damit eine Berechnung auf

einem klassischen Rechner möglich wäre.

Komplexität und Informationsverlust

Ohne den Effizienzgewinn zu verlieren ist eine exakte Bestimmung des Ergebnisses, den der Quantenalgorithmus liefert, nicht möglich. Andererseits kann der Rechenaufwand nur unter immensem Informationsverlust auf das gewünschte Niveau gebracht werden. Eine ausreichend gute Annäherung an das exakte Ergebnis bei einer immer noch signifikanten Einsparung von Berechnungsschritten auf einem klassischen Rechner könnte einen interessanten Mittelweg darstellen. Eine signifikante Einsparung heißt dabei, möglichst in einer niedrigeren Ordnungsklasse der Komplexität zu landen, als dies im günstigsten Fall bei einer exakten Berechnung auf einem Rechner möglich wäre. Das approximierte Ergebnis sollte dann bei einer mehrfach durchgeführten Simulation ein dem tatsächlichen Ergebnis sehr angenähertes Ergebnis liefern.

Quantenalgorithmen als stochastische Algorithmen

Die Komplexität bei der Berechnung des exakten Ergebnisses eines Quantenalgorithmus ist zu groß, als dass es sich effizient auf einem klassischen Rechner umsetzen ließe. Die aus der Physik stammenden Algorithmen weisen ein Verhalten auf, das von stochastischer Natur ist und zu einem Vergleich mit stochastischen Verfahren ermuntert. Deswegen wird ein auf einem stochastischen Algorithmus beruhendes Verfahren konstruiert, bei dem die Reduktion der Komplexität auf ein niedrigeres Niveau Ansatzpunkte für die klassische Umsetzung bringt. Die Studie, wie das mit dem speziellen Branch&Bound-Verfahren gemacht wird, stellt den Kern der Untersuchung dar. Das Niveau der Komplexität wird zunächst auf dieselbe Ordnung gebracht, die der Quantenalgorithmus bei der Ausführung auf einem Quantenrechner benötigt. Dies ist jedoch mit einem großen Informationsverlust verbunden. Wird nach und nach der Rechenaufwand erhöht, gewinnt man zusätzliche Information. Das Abwägen zwischen möglichst geringem Informationsverlust und einer tragbaren Ordnung der Komplexität ist das entscheidende Kriterium bei der klassischen Simulation von Quantenalgorithmen. Abbildung 1 zeigt schematisch den gewünschten Verlauf zwischen der Komplexität und dem Informationsverlust.

Aufbau der Arbeit

Kapitel 1

Nach einer fundierten Betrachtung der zugrunde liegenden Theorie kann ein Verfahren, das den Rechenaufwand bei gleichzeitigem Informationsverlust deutlich reduziert, entwickelt werden. Deswegen steht eine kurze Rekapitulation des Begriffs „Quantum Computation“ (QC) am Anfang der Arbeit. Die Quantenmechanik (QM) ist dafür die Grundlage und deshalb müssen zunächst die entscheidenden Pfeiler zur Begründung des QC aus der QM gelegt werden, um daraus die Begriffe „Multi-q-Bit“ und „Gate“ entwickeln zu können.

Kapitel 2

Aufbauend auf die einführenden Abschnitte steht die unitäre Darstellung der diskretisierten zeitlichen Entwicklung eines Systems im Mittelpunkt. Eine solche zeitliche Entwicklung lässt sich über spezielle Transformationsmatrizen, die stets lediglich auf einem Bit arbeiten, konstruieren. Ist

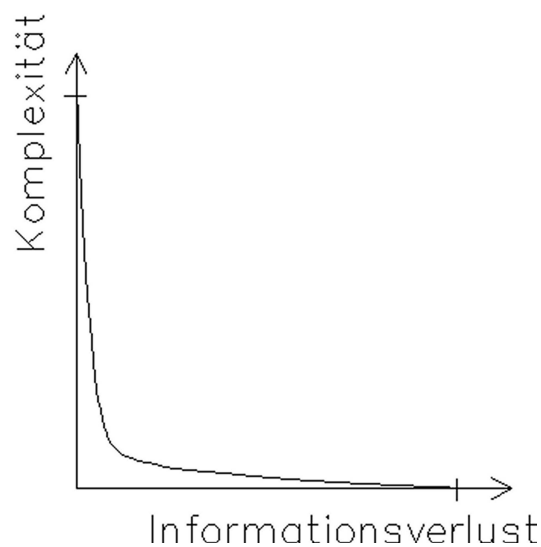


Bild 1: Komplexität vs. Informationsverlust

umgekehrt eine zeitliche Entwicklung gegeben, kann diese als Abfolge von Ein-Bit-Operationen interpretiert werden und entsprechend zerlegt werden. Die besondere Festlegung dieser Operationen lässt eine veränderte Darstellung der zeitlichen Entwicklung in Form von Ein-Bit-Operationen zu. Aus solchen Ein-Bit-Operationen wurden und werden die vielfältigsten Algorithmen entwickelt.

Kapitel 3

QA zeichnen sich durch drei bestimmte Charakteristika aus. Erstens muss das System in einen festgelegten Anfangszustand gebracht werden, zweitens vollzieht sich die zeitliche Entwicklung und drittens wird eine Messung eines oder mehrerer Bits durchgeführt. Zusätzlich muss jedoch zwischen einem zeitkontinuierlichen und einem zeitdiskreten Modell unterschieden werden. Die Betrachtungen in dieser Arbeit benötigen das diskrete Modell. Deswegen müssen verschiedene Techniken erwähnt werden, um vom kontinuierlichen zum diskreten Modell zu kommen. Es gibt verschiedene Klassen von QA. [Lomo04@] nennt gerade einmal zwei allgemeine Klassen von QA: Die Hidden Subgroup-Probleme und Algorithmen zur Amplitudenverstärkung. Eine dritte von ihm noch unerwähnte Klasse soll hier hinzugefügt werden: Quantum-(Random)-Walks. Sie sollen in eigenen Unterabschnitten dargestellt werden.

Kapitel 4

Es wird der Ansatz der Approximation eines zeitdiskreten QA basierend auf einem stochastischen Algorithmus gezeigt. Ein QA kann ohne die Betrachtung einer Messung als Binärbaum dargestellt werden. Die Tiefe des Baumes entspricht dem zeitlichen Fortschritt. Wird die Komplexität derart reduziert, dass auf stochastischer Grundlage vom Ausgangsknoten hinab zu den untersten Knoten (den Blättern) des Baumes genau ein Pfad durchlaufen wird, kann ein stochastischer Algorithmus formuliert werden. Wichtig dabei sind die Güte und die Abschätzung des Aufwandes eines solchen Verfahrens. Jeder Repräsentant einer Klasse von QA eignet sich in unterschiedlicher Weise für die Berechnung mit dem vorgestellten Verfahren.

Kapitel 5

Wird der Algorithmus implementiert, können verschiedene Informationen im Hinblick auf die Reduktion des Berechnungsaufwandes und den Verlauf zwischen Informationsverlust und Komplexität gewonnen werden. Die Ergebnisse der durchgeführten Simulationen werden im vorletzten Kapitel dargestellt. Zur Simulation ist ein auf JAVA basierendes Tool implementiert und eingesetzt. Es werden der Suchalgorithmus von Grover und ein Quantum Walk-Algorithmus numerisch und grafisch untersucht.

Kapitel 6

Ein zusammenfassender Abschnitt mit einem kurzen Blick in die Perspektiven solcher approximierender Verfahren bildet den Abschluss dieser Arbeit. Die Studie zur Simulation von QA auf klassischen Rechnern zeigt die Möglichkeiten aber auch die Schwierigkeiten beim Finden der richtigen Balance zwischen vertretbarem Informationsverlust und der notwendigen Komplexität bei der Approximation von QA auf klassischen Rechnern.

KAPITEL 1

Quantenmechanische Grundlagen

Die einem Quantencomputer zugrunde liegende Theorie beruht auf physikalischen Zusammenhängen der Quantenmechanik. Dabei müssen im Gegensatz zur klassischen Mechanik andere und oft sogar über die Vorstellungskraft hinausreichende Zusammenhänge angenommen werden. Zudem sind die grundsätzlichen Zusammenhänge axiomatisch zu verstehen. Beispielsweise kann eine Schrödinger-Gleichung nicht mathematisch bewiesen werden. Diese „Postulate“, die den Zustandsraum (Abschnitt 1.1), die messbaren Größen (Abschnitt 1.2) und die zeitliche Entwicklung (Abschnitt 1.3) beschreiben, bilden aber die Grundlage für die darauf aufgesetzte Theorie des QC.

1.1 Physikalischer Zustandsraum

Im mikroskopisch kleinen Bereich der Quantenmechanik steht die Eigenschaft des **Welle-Teilchen-Dualismus** im Vordergrund. Licht etwa besitzt Teilcheneigenschaften¹, die sich in Form von Energiequanten und einem konkreten Impuls verifizieren lassen. Dementsprechend besitzen Teilchen Welleneigenschaften² in Form von Beugungs- und Interferenzeigenschaften. Ein jedes Teilchen kann so durch eine **Wellenfunktion** vollständig charakterisiert werden (vgl. [Müth99]). Eine physikalische Anordnung, die mit einer Wellenfunktion beschrieben werden kann, befindet sich zu einem festen Zeitpunkt in einem „stationären Zustand“ [Müth99].

Ein zusätzlicher Parameter der Wellenfunktion ist der **Teilchen-Spin** σ , der unabhängig von den übrigen Parametern ist. Es gibt zwei Möglichkeiten für den Spin: Spin-up (\uparrow) und Spin-down (\downarrow).

Mathematisch gesehen ist eine Wellenfunktion zunächst eine Abbildung von einem k -dimensionalen reellen Parameterraum in die komplexen Zahlen.

$$\psi = \psi_\sigma(\vec{x}), \quad \psi : \mathbb{R}^k \rightarrow \mathbb{C}$$

Die Menge der Wellenfunktionen zur Beschreibung eines Teilchens lässt sich zu einem Funktionenraum zusammenfassen

$$L^2(\mathbb{R}^k) \stackrel{def}{=} \{\psi : \mathbb{R}^k \rightarrow \mathbb{C} \mid \psi \text{ messbar und } |\psi|^2 \text{ integrierbar}\}$$

und $L^2(\mathbb{R}^k)$ ist damit gemäß [Aulb96] ein komplexer unendlichdimensionaler Vektorraum. Dazu ist zu sagen, dass bestimmte Wellenfunktionen hinsichtlich Messbarkeit und Integrierbarkeit nicht

¹„de Broglie-Beziehung“: Einer ebenen Welle kann eine gleichförmige Bewegung mit einer bestimmten Energie zugeordnet werden.

²„Einstein’sche Beziehung“: Materieteilchen entsprechen Materiewellen mit einer bestimmten Kreisfrequenz.

zu unterscheiden sind und deswegen die Äquivalenzrelation

$$\psi_1 \sim \psi_2 \stackrel{\text{def}}{\Leftrightarrow} \psi_1(\vec{x}) = \psi_2(\vec{x}) \quad \text{für fast alle } \vec{x} \in \mathbb{R}^k$$

betrachtet wird. Es wird die Faktorisierung $\mathcal{L}^2(\mathbb{R}^k) \stackrel{\text{def}}{=} L^2(\mathbb{R}^k)/N$ von $L^2(\mathbb{R}^k)$ nach dem Unterraum $N \stackrel{\text{def}}{=} \{\psi \in L^2(\mathbb{R}^k) \mid \int_{\mathbb{R}^k} |\psi(\vec{x})|^2 d\vec{x} = 0\}$ betrachtet. Die Festlegung bildet einen mit einem Skalarprodukt und einer dazu korrespondierenden Normeigenschaft ($\|\psi\|_2 = \langle \psi, \psi \rangle^{\frac{1}{2}}$) für die beschriebenen Wellenfunktionen versehenen Hilbertraum (vgl. Definition 1 und [Münn02@]). Ein Element aus $\mathcal{L}^2(\mathbb{R}^k)$ entspricht dabei einem **Zustand** des Systems.

Es gibt einen ganzen (Vektor-)Strahl aus Funktionswerten einer Wellenfunktion, der denselben Zustand beschreibt. Durch eine Normierungsforderung werden äquivalente Zustandsbeschreibungen auf längentreue Darstellungen eingeschränkt. Lediglich durch Phasenverschiebungen ist eine Beschreibung gleicher Zustände noch möglich. Damit kann bei Normierung auf 1 von einer Wahrscheinlichkeit, ein Teilchen an einer bestimmten Stelle nachzuweisen, gesprochen werden. Diese ist durch das Betragsquadrat der Wellenfunktion gegeben, durch die das Teilchen beschrieben wird (vgl. [Müth99]).

Das physikalische System kann verschiedene Zustände einnehmen, die durch jeweils eine Wellenfunktion ψ ausgedrückt werden und wegen der Linearität des Hilbertraumes kann es auch alle durch eine Linearkombination dieser Zustände entstehenden Zustände einnehmen ([Bouw00]).

Satz 1.1 (Superpositionsprinzip) Seien ψ_k , $k = 1, \dots, m$ Zustände eines Hilbertraumes $\mathcal{L}^2(\mathbb{R}^k)$. Dann entsteht ein anderer Zustand $\psi \in \mathcal{L}^2(\mathbb{R}^k)$ durch

$$\psi = \sum_{k=1}^m \alpha_k \psi_k$$

für beliebige $\alpha_k \in \mathbb{C}$.

Beweis. Das folgt unmittelbar aus den Vektorraumeigenschaften. □

Bemerkung. Eine Linearkombination ist nicht unbedingt längenerhaltend, selbst wenn eine Normierungsbedingung gilt. Der neu entstandene Zustand befindet sich aber auf einem Vektorstrahl, auf dem sich auch äquivalente normierte Zustände befinden. Hierauf wird in Abschnitt 2.1 noch näher eingegangen.

In der Quanteninformationstheorie ist der Teilchen-Spin der einzig betrachtete Parameter. Denn damit kann eine Identifizierung des Spins mit der klassischen Informationseinheit, dem **Bit** $b \in B$, $|B| = 2$, erfolgen. Spin-up entspricht dann beispielsweise „Bit gesetzt“ und Spin-down „Bit nicht gesetzt“. Die auf die Betrachtung des Spins fokussierte Wellenfunktion wird **Spinor-Wellenfunktion** genannt (vgl. [Münn02@]). Die Unabhängigkeit des Spins von den anderen

Parametern ermöglicht die Aufteilung der Wellenfunktion in zwei Wellenfunktionen, die zu

$$\psi_\sigma(\vec{x}) = \begin{pmatrix} \psi_\uparrow(\vec{x}) \\ \psi_\downarrow(\vec{x}) \end{pmatrix} \quad (1.1)$$

zusammengefasst werden. Die Wellenfunktion ist dann eine 2-Komponenten-Funktion mit der Spin-up- und der Spin-down-Komponente. Insgesamt ergibt sich das erste Postulat zur Beschreibung eines physikalischen Systems.

Postulat 1 (Physikalischer Zustandsraum) *Der (stationäre) Spin-Zustand eines physikalischen Systems wird durch eine quadratintegrale **Spinor-Wellenfunktion***

$$\psi = \psi_\sigma(\vec{x}) = \begin{pmatrix} \psi_\uparrow(\vec{x}) \\ \psi_\downarrow(\vec{x}) \end{pmatrix}, \quad \psi : \mathbb{R}^k \rightarrow \mathbb{C}^2$$

im **Hilbertraum** $\mathcal{L}^2(\mathbb{R}^k)$ über \mathbb{C} in Abhängigkeit des Parameters $\vec{x} = (x_1, \dots, x_k)$, $x_i \in \mathbb{R} \forall i = 1, \dots, k$ charakterisiert. Zudem soll die Normierungsbedingung

$$\int_{\mathbb{R}^k} |\psi_\sigma(\vec{x})|^2 d\vec{x} = \int_{\mathbb{R}^k} |\psi_\uparrow(\vec{x})|^2 d\vec{x} + \int_{\mathbb{R}^k} |\psi_\downarrow(\vec{x})|^2 d\vec{x} = 1 \quad (1.2)$$

gelten.

Vektorielle Sichtweise

Der vom Zufall beeinflusste Spin-Zustand eines Teilchens kann als Zufallsexperiment

$$(\{\uparrow, \downarrow\}, \{\{\emptyset\}, \{\uparrow, \downarrow\}, \{\uparrow\}, \{\downarrow\}\}, P) \quad (1.3)$$

beschrieben werden. Wird ein Teilchen betrachtet, so ist die Wahrscheinlichkeit dafür, dass es ein Spin-up-Teilchen ist, gleich dem Ausdruck

$$P(\uparrow) = \int_{\mathbb{R}^k} |\psi_\uparrow(\vec{x})|^2 d\vec{x}$$

in Gleichung (1.2). Analog wird die Wahrscheinlichkeit für ein Spin-down-Teilchen bestimmt.

Bemerkung. Die Spinor-Wellenfunktion liefert einen Ergebnisvektor in \mathbb{C}^2 . Da bei der Betrachtung der Spinor-Wellenfunktion lediglich interessant ist, ob es sich um ein Spin-up- oder ein Spin-down-Teilchen handelt, lässt sich vereinfacht schreiben

$$\psi = \begin{pmatrix} \psi_\uparrow \\ \psi_\downarrow \end{pmatrix}, \quad P(\uparrow) = |\psi_\uparrow|^2, \quad P(\downarrow) = |\psi_\downarrow|^2$$

Die möglichen Zustände eines solchen Spin-Teilchens werden vollständig durch die Vektoren in einem 2-dimensionalen Hilbertraum mit diskreter Basis beschrieben. Deswegen bezeichne von nun an \mathcal{H} einen Hilbertraum der Dimension 2 über \mathbb{C} . \mathcal{H} ist der Zustandsraum eines Spin-Teilchens. Jeder Vektor in \mathcal{H} „entspricht“ dabei Funktionswerten einer oder mehrerer Spinor-Wellenfunktionen in $\mathcal{L}^2(\mathbb{R}^k)$. Die nachfolgenden Ausführungen für jedes Spin-Teilchen (bzw. „Bit“) beziehen sich stets auf die vektorielle Sichtweise im Hilbertraum \mathcal{H} .

Vektoren aus dem endlichen Hilbertraum \mathcal{H} werden in der Dirac-Notation als Spaltenvektoren mit $\psi = |\psi\rangle$ geschrieben. Sie heißen **ket-Vektoren**. Ihr konjugiert-komplexes Gegenstück, die **bra-Vektoren**, sind Zeilenvektoren und werden $\psi^\dagger = \langle\psi|$ geschrieben. Damit ergibt sich das bekannte Skalarprodukt in \mathcal{H} , $\langle\psi|\psi\rangle_{\mathcal{H}} = \langle\psi|\psi\rangle$. Es entsteht so der zur Dirac-Notation synonyme Begriff „bracket“-Notation.

Sei in \mathcal{H} durch

$$B = \{b_1, b_2 : b_1, b_2 \in \mathcal{H}\}$$

eine Orthonormalbasis von \mathcal{H} ausgezeichnet. Dabei sei für einen Zustand $\psi \in \mathcal{H}$ $a_k = \langle b_k|\psi\rangle$. Jedes ψ zu einer beliebigen Basis B kann als Superposition aus Satz 1.1 der Basiszustände dargestellt werden. Wird zudem

$$|a_1|^2 + |a_2|^2 = 1 \quad (1.4)$$

gefordert, ist die Superposition längenerhaltend und die Normierungsbedingung aus Postulat 1 erfüllt.

Die Erweiterung auf eine beliebige Anzahl von n Spin-Teilchen wird mit Hilfe des Tensorproduktes ermöglicht.

$$\mathcal{H}^{\otimes n} = \bigotimes_n \mathcal{H}$$

Dabei ist $\mathcal{H}^{\otimes n}$ ein zu einem Hilbertraum der Dimension 2^n über \mathbb{C} isomorpher Vektorraum. Jeder Zustand ψ aus $\mathcal{H}^{\otimes n}$ kann geschrieben werden als

$$\psi = \sum_{i_1=1}^2 \dots \sum_{i_n=1}^2 a_{i_1 \dots i_n} b_{i_1} \otimes \dots \otimes b_{i_n}$$

Das Tensor-Produkt dieser Vektoren wird in der Dirac-Notation geschrieben als

$$\psi \stackrel{def}{=} |\psi_1 \dots \psi_n\rangle = |\psi_1\rangle \dots |\psi_n\rangle = |\psi_1\rangle \otimes \dots \otimes |\psi_n\rangle \in \mathcal{H}_1 \otimes \dots \otimes \mathcal{H}_n.$$

Definition 1.2 (Reine Zustände) In einem Hilbertraum $\mathcal{H}^{\otimes n}$ der Dimension 2^n sei eine Orthonormalbasis $B = \{b_1, \dots, b_{2^n} : b_i \in \mathcal{H}^{\otimes n}\}$ ausgezeichnet. Dann heißt $\psi \in B$ **reiner Zustand** in $\mathcal{H}^{\otimes n}$.

Mit den Basisvektoren b_k einer Orthonormalbasis lassen sich beliebige Zustände durch Linearkombination mit Skalaren a_k bilden: $|\psi\rangle = \sum_{k=1}^{2^n} a_k |b_k\rangle$.

1.2 Operatoren

Zustände können auf andere Zustände abgebildet werden. Solche Abbildungen von Teilräumen eines Hilbertraumes \mathcal{H} nach \mathcal{H} heißen Operatoren. Insbesondere interessant sind dabei lineare Operatoren, da sie das Prinzip der Superposition nicht verletzen.

Definition 1.3 ((Lineare) Operatoren) *Eine Abbildung*

$$\begin{aligned} A : \mathcal{D}_A &\rightarrow \mathcal{H}, \quad \mathcal{D}_A \subset \mathcal{H} \\ \psi &\mapsto A\psi \end{aligned}$$

heißt **Operator** von einem Teilraum \mathcal{D}_A nach \mathcal{H} . A heißt **linear**, falls gilt

$$A(\alpha_1\psi_1 + \alpha_2\psi_2) = \alpha_1 A\psi_1 + \alpha_2 A\psi_2$$

Bestimmte Operatoren sind von stärkerer Bedeutung und sollen deshalb erwähnt werden. Sei A ein Operator. Der zu A **adjungierte** Operator A^\dagger ist gekennzeichnet durch

$$\langle \psi_1 | A\psi_2 \rangle_{\mathbb{C}} = \langle A^\dagger\psi_1 | \psi_2 \rangle_{\mathbb{C}}, \quad \forall \psi_1 \in \mathcal{D}_{A^\dagger}, \psi_2 \in \mathcal{D}_A.$$

Darüber hinaus ist A **hermitesch**, wenn gilt

$$\langle \psi_1 | A\psi_2 \rangle_{\mathbb{C}} = \langle A\psi_1 | \psi_2 \rangle_{\mathbb{C}}, \quad \forall \psi_1, \psi_2 \in \mathcal{D}_A, \mathcal{D}_{A^\dagger} \subseteq \mathcal{D}_A.$$

A heißt **selbstadjungiert**, falls $A = A^\dagger$ und $\mathcal{D}_{A^\dagger} = \mathcal{D}_A$. Damit ist ein selbstadjungierter Operator auch hermitesch. Eigenwerte spielen bei physikalischen Systemen eine wesentliche Rolle. Dabei ist ein **Eigenwert** $a \in \mathbb{C}$ eines Operators A zum **Eigenzustand** $\psi \in \mathcal{H}$ definiert durch

$$A\psi = a\psi. \tag{1.5}$$

Satz 1.4 (Eigenwerte und -vektoren hermitescher Operatoren) *Sei A ein hermitescher Operator. Dann sind die Eigenwerte von A reell und die Eigenvektoren von A zu verschiedenen Eigenwerten sind orthogonal.*

Beweis. Sei A hermitesch.

Teil I:

$$\begin{aligned} A\psi = a\psi &\Rightarrow \langle \psi | A\psi \rangle_{\mathbb{C}} = \langle \psi | a\psi \rangle_{\mathbb{C}} = \bar{a} \langle \psi | \psi \rangle_{\mathbb{C}} \\ \langle \psi | A\psi \rangle_{\mathbb{C}} &= \langle A\psi | \psi \rangle_{\mathbb{C}} = \langle a\psi | \psi \rangle_{\mathbb{C}} = a \langle \psi | \psi \rangle_{\mathbb{C}} \\ &\Rightarrow a = \bar{a} \end{aligned}$$

Teil II: Sei $A\psi_1 = a_1\psi_1$, $A\psi_2 = a_2\psi_2$, $a_1 \neq a_2$.

$$\left. \begin{aligned} \langle \psi_1 | A\psi_2 \rangle_{\mathbb{C}} &= a_2 \langle \psi_1 | \psi_2 \rangle_{\mathbb{C}} \\ \langle A\psi_1 | \psi_2 \rangle_{\mathbb{C}} &= a_1 \langle \psi_1 | \psi_2 \rangle_{\mathbb{C}} \end{aligned} \right\} \Rightarrow (a_2 - a_1) \langle \psi_1 | \psi_2 \rangle_{\mathbb{C}} = 0 \Rightarrow \langle \psi_1 | \psi_2 \rangle_{\mathbb{C}} = 0$$

□

Observablen

Die (physikalischen) Messgrößen eines physikalischen Systems werden **Observablen** genannt. Sie können zu einem festen Zeitpunkt und während eines bestimmten Zustandes gemessen werden. In der Quantenmechanik ist festgelegt, dass Observablen als lineare Operatoren auf den Wellenfunktionen wirken. Zudem besitzen die möglichen Messwerte einer Observablen eine Verteilung. Da Messwerte reelle Zahlen sein müssen (um gemessen werden zu können) und die Eigenwerte der Operatoren die einzig messbaren Werte sind (vgl. [Müth99], [Heis79]), werden den Observablen selbstadjungierte Operatoren zugeordnet. Nach einer Messung findet eine **Zustandsreduktion** statt. Das bedeutet, dass das System danach in den zum gemessenen Eigenwert a des Operators A gehörenden Eigenzustand übergeht (vgl. [Münn02@]): Eigenzustand ψ mit $A\psi = a\psi$ wird Systemzustand.

Der Erwartungswert einer Observablen A im Zustand ψ (Eigenwert: a) ist gegeben durch

$$\langle A \rangle = \langle \psi | A | \psi \rangle \stackrel{def}{=} \langle \psi | A\psi \rangle_{\mathbb{C}} = \langle \psi | a\psi \rangle_{\mathbb{C}} = a \langle \psi | \psi \rangle_{\mathbb{C}} = a.$$

Die Dirac-Notation für die Anwendung von Observablen auf Systemzustände ist analog: $A\psi = A|\psi\rangle$ bzw. $\psi^\dagger A = \langle \psi | A$.

Zwei Observablen A_1 und A_2 heißen **verträglich**, wenn gilt $A_1A_2 - A_2A_1 = 0$. Deshalb wird der **Kommutator** zweier Observablen A_1 und A_2 definiert als

$$[A_1, A_2] \stackrel{def}{=} A_1A_2 - A_2A_1 \quad (1.6)$$

Die Verträglichkeit wird noch eine wichtige Rolle spielen. Observablen sind im Allgemeinen nicht verträglich. Die Nicht-Verträglichkeit von Observablen hat nämlich zur Folge, dass die betroffenen Operationen nicht gleichzeitig scharf messbar sind. Es gibt stattdessen eine Unschärfesituation. Die Unschärfe ist einer der Eckpunkte der Kopenhagener Deutung (siehe [Heis79]) der Quantentheorie, auf die kurz in Abschnitt 1.4 eingegangen wird. Zusammenfassend lautet das zweite Postulat der Quantenmechanik somit:

Postulat 2 (Observablen und Messung) *Den **Observablen** eines physikalischen Systems entsprechen selbstadjungierte Operatoren. Alle möglichen Messergebnisse sind die (reellen) Eigenwerte des Operators. Der Erwartungswert einer Observablen A im Zustand ψ lautet $\langle \psi | A | \psi \rangle$. Nach einer **Messung** befindet sich das System in einem Eigenzustand bezüglich A und dem gemessenen Eigenwert.*

Bisher stand ein stationärer Zustand im Vordergrund. Nun soll sich ein System aber in der Zeit verändern können. Der folgende Abschnitt betrachtet diese zeitliche Entwicklung des Systems.

1.3 Zeitliche Dynamik des Systems

Eine Differentialgleichung erster Ordnung beschreibt die Entwicklung des Systems in Abhängigkeit der Zeit t . Die **zeitabhängige Schrödinger-Gleichung**, welche die zeitlich-dynamische Entwicklung des Systems beschreibt, lautet³

$$i\hbar \frac{\partial \psi(t)}{\partial t} = H \psi(t) \quad (1.7)$$

H ist der **Hamiltonoperator**⁴ des Systems. Für eine sinnvolle physikalische Anordnung ist der Hamiltonoperator selbstadjungiert zu wählen. Denn nur so werden reelle Eigenwerte und damit tatsächlich messbare Energiezustände des Systems erreicht (gemäß Postulat 2). Ist das System in einem Anfangszustand

$$\psi(0) = \psi_0$$

festgelegt und hängt der Hamiltonoperator nicht von der Zeit t ab, also

$$\frac{\partial H}{\partial t} = 0, \quad (1.8)$$

kann die Gleichung (1.7) gelöst werden:

Satz 1.5 (Lösung der zeitabhängigen Schrödinger-Gleichung) *Bei gegebenem zeitunabhängigen Hamiltonoperator hat die zeitabhängige Schrödinger-Gleichung die Lösung*

$$\psi(t) = e^{-\frac{iHt}{\hbar}} \psi_0 \quad (1.9)$$

Beweis durch Einsetzen in Gleichung (1.7).

□

Die gewonnene Lösung der zeitabhängigen Schrödinger-Gleichung ist der **Zeitentwicklungsoperator**

$$U(t) \stackrel{\text{def}}{=} e^{-\frac{iHt}{\hbar}} \quad (1.10)$$

des Systems.

Unitarität

Wird nun ein selbstadjungierter zeitunabhängiger Hamiltonoperator H als Generator gewählt, ist die Zeitentwicklung unitär. Das System heißt dann **konservativ**. Unitär bedeutet hierbei Folgendes: Seien $M, N \in \mathcal{H}^{k,k}$, $k \in \mathbb{N}$ Matrizen und $\psi \in \mathcal{H}^k$ ein Vektor. Dann gilt:

M ist unitär \Leftrightarrow Die Spalten von M sind normiert und

³Die Naturkonstante \hbar hat den Wert $\hbar = \frac{h}{2\pi}$, wobei h das Planck'sche Wirkungsquantum ist mit $h = 6.6260755 \cdot 10^{-34} \text{Js}$ (vgl. [Müth99])

⁴Der Hamiltonoperator ist ein grundlegender Operator der Quantentheorie, der auf die möglichen Zustandsvektoren des betrachteten mikrophysikalischen Systems wirkt und (...) damit ihre zeitlichen Veränderungen festlegt. [Meye73]

paarweise orthogonal.

$$M \cdot M^\dagger = M^\dagger \cdot M = \mathbb{1}_k$$

$$M^{-1} = M^\dagger \quad (1.11)$$

M ist unitär $\Rightarrow M^\dagger$ ist unitär

$$|\det M| = 1$$

$$\|M\psi\| = \|\psi\|$$

$$|\lambda| = 1 \quad \text{für alle Eigenwerte } \lambda \text{ von } M \quad (1.12)$$

$$M, N \text{ sind unitär} \Rightarrow M \cdot N \text{ bzw. } N \cdot M \text{ sind unitär} \quad (1.13)$$

Aber: $[M, N] \neq 0$ i.A. vgl. (1.6)

Satz 1.6 (Unitarität der Zeitentwicklung) *Ist H in (1.7) selbstadjungiert und zeitunabhängig, so ist die Zeitentwicklung des Systems **unitär**.*

Beweis. Sei H selbstadjungiert. Zu zeigen ist, dass $U(t) \cdot U^\dagger(t) = \mathbb{1}_k$ (in der Dimension k) gilt. Da H hermitesch ist, ist H gemäß [Råde97] diagonalisierbar mit einer unitären Matrix \bar{U} und den nicht notwendig verschiedenen reellen Eigenwerten a_1, \dots, a_k :

$$\bar{U}^\dagger H \bar{U} = \text{diag}(a_1, \dots, a_k) \stackrel{\text{def}}{=} D.$$

Damit gilt dann

$$\begin{aligned} U(t) \cdot U^\dagger(t) &= e^{-\frac{iHt}{\hbar}} \cdot e^{\frac{iHt}{\hbar}} = e^{-\frac{i\bar{U}D\bar{U}^\dagger t}{\hbar}} \cdot e^{\frac{i\bar{U}D\bar{U}^\dagger t}{\hbar}} \\ &\stackrel{(3.12)}{=} \bar{U} e^{-\frac{iDt}{\hbar}} \bar{U}^\dagger \cdot \bar{U} e^{\frac{iDt}{\hbar}} \bar{U}^\dagger \\ &= \bar{U} e^{-\frac{it}{\hbar}D} \cdot e^{\frac{it}{\hbar}D} \bar{U}^\dagger \\ &= \bar{U} \cdot \text{diag}(e^{-\frac{ia_1 t}{\hbar}}, \dots, e^{-\frac{ia_k t}{\hbar}}) \cdot \text{diag}(e^{\frac{ia_1 t}{\hbar}}, \dots, e^{\frac{ia_k t}{\hbar}}) \cdot \bar{U}^\dagger \\ &= \mathbb{1}_k \end{aligned}$$

□

Es ergibt sich das dritte Postulat der Quantenmechanik.

Postulat 3 (Zeitentwicklung des physikalischen Systems) Die zeitabhängige Entwicklung der Zustände eines physikalischen Systems wird durch die **zeitabhängige Schrödinger-Gleichung**

$$i\hbar \frac{\partial \psi(t)}{\partial t} = H \psi(t) \quad (1.14)$$

beschrieben. Der bei einem selbstadjungierten und zeitunabhängigen Hamiltonoperator resultierende **Zeitentwicklungsoperator**

$$U(t) = e^{-\frac{iHt}{\hbar}} \quad (1.15)$$

ist unitär.

1.4 Zusammenfassung

Im Frühjahr des Jahres 1927 gelangte man zu einer widerspruchsfreien Deutung der Quantentheorie, die als **Kopenhagener Deutung** bekannt ist (siehe [Heis79, Bouw00]). Die Durchführung eines Experimentes erfordert drei Schritte: Die Startanordnung muss in ein Wahrscheinlichkeitsmaß für den Systemzustand umgesetzt werden. Die Zeitentwicklung des Systems als zweiter Schritt verändert dieses Maß im Laufe der Zeit. Der dritte Schritt ist dann die Messung des Systems. Die Wahrscheinlichkeit, dass bei einer Messung ein bestimmter Zustand gemessen wird, ist gleich dem Quadrat des Betrages des Funktionswertes der Wellenfunktion bezüglich dieses Zustandes.

Eine weitere Interpretation in [Heis79] erklärt den mikroskopischen und kontraintuitiven Charakter der Quantentheorie. Gegebenheiten, die sich mathematisch gesehen im makroskopischen Bereich befinden (etwa alle Abläufe aus unserer alltäglichen Sicht), d.h., bei denen das Planck'sche Wirkungsquantum eine vergleichsweise kleine Größe darstellt, verlaufen nahezu gemäß der klassischen Gesetze. Je größer der Einfluss der Naturkonstante wird, desto mehr kommt die Wahrscheinlichkeitsinterpretation zum Tragen und die quantentheoretischen Effekte beginnen zu wirken.

Die Quantenmechanik liefert die Grundlagen zum QC. Ein physikalisches System lässt sich in die Struktur eines Hilbertraumes einbinden. Physikalisch gesehen wird der Zustand eines Spin-Teilchens über eine Spinor-Wellenfunktion aus dem Hilbertraum beschrieben. Die Spinor-Wellenfunktion liefert ein Wahrscheinlichkeitsmaß für das Spin-Teilchen. Äquivalent dazu wird das Wahrscheinlichkeitsmaß auch über den Bildvektor der Spinor-Wellenfunktion festgelegt. Der Bildvektor ist ein komplexer Vektor aus \mathbb{C}^2 . Die Bildung des Tensorproduktes wird zur Beschreibung eines Systems aus n Spin-Teilchen benötigt. Der dazugehörige Vektorraum ist ein Hilbertraum der Dimension 2^n über \mathbb{C} , der zu dem Vektorraum \mathbb{C}^{2^n} isomorph ist. Ein Zustand kann aus anderen Zuständen über das Superpositionsprinzip erzeugt werden.

Die Kopenhagener Deutung zeigt, dass das tatsächliche Einnehmen eines Zustandes nicht vorhersagbar ist, sondern es kann nur die Wahrscheinlichkeit für ein Ergebnis der Messung vorhergesagt werden. Durch mehrmalige Wiederholung eines Experimentes kann ein solches Ereignis nachgeprüft werden. Ohne eine Messung ist damit der Zustand des Systems indefinit. Eine physikalische Anordnung liefert reelle und damit messbare Ergebnisse in Form von Eigenwerten. Deswegen sind selbstadjungierte Operatoren erforderlich. Nach einer Messung findet eine Systemreduktion statt und das System geht in den Eigenzustand bezüglich des gemessenen Eigenwertes über.

Die zeitliche Entwicklung des Systems über Operatoren, die unter bestimmten Voraussetzungen unitär ist, ändert die Wahrscheinlichkeiten der Zustände. Dadurch ist die Möglichkeit der nun folgenden Einführung des QC aus dem Bereich der Informationstheorie und der damit einhergehenden algorithmischen Nutzung der Theorie gegeben. In [Bouw00] wird es so formuliert: Es „ist zu ersehen, dass das ganze mathematische Rüstzeug der Quantenmechanik in der Sprache von Operatoren in Matrixform und unitären Transformationen formuliert werden kann“.

KAPITEL 2

Quantum Computation

Der Begriff des Quantum Computation verbindet die beiden Gebiete der Quantentheorie und der Informatik. Die Informatik nutzt die Quantenmechanik zur Formulierung einer Darstellung zur Rechnung in Bits, wie sie durch die heutige Computertechnik geläufig ist. Zu diesem Zweck sind verschiedene Begriffe einzuführen. Die physikalischen Vorgaben müssen modelliert werden (Abschnitt 2.1). Die zeitliche Entwicklung kann einerseits aus einfach gehaltenen Einzelschritten konstruiert werden (Abschnitt 2.2). Andererseits ist es möglich, eine vorgegebenen zeitliche Entwicklung in einfache Einzelschritte aufzuschlüsseln (Abschnitt 2.3).

2.1 Mathematische Modellierung

Zunächst steht ein stationärer Zustand ohne zeitliche Dynamik im Mittelpunkt. Die beiden ersten Postulate aus der Quantenmechanik sollen umgesetzt werden.

2.1.1 Mathematischer Zustandsraum

Jeder Zustand eines physikalischen Systems wird durch einen komplexen Vektor aus dem Hilbertraum \mathcal{H} repräsentiert. Die einzelnen Vektoren lassen sich klassifizieren.

Satz 2.1 *Auf der Menge der Vektoren aus dem Hilbertraum \mathcal{H} über \mathbb{C} ist eine Äquivalenzrelation R gegeben mit $\psi_1, \psi_2 \in \mathcal{H}$, $\psi_k \neq 0$ und*

$$\psi_1 \sim \psi_2 \iff \psi_1 = \lambda \psi_2 \quad \text{für ein } \lambda \in \mathbb{C}$$

Beweis. Seien $\psi_1, \psi_2, \psi_3 \in \mathcal{H}$, $\psi_k \neq 0$, $\lambda, \mu \in \mathbb{C}$, $\lambda, \mu \neq 0$.

Reflexivität: $\psi_1 \sim \psi_1$: $\psi_1 = 1 \cdot \psi_1$, $1 \in \mathbb{C}$.

Symmetrie: $\psi_1 \sim \psi_2$: $\psi_1 = \lambda \psi_2 \implies \psi_2 = \frac{1}{\lambda} \psi_1$, $\frac{1}{\lambda} \in \mathbb{C}$, $\psi_2 \sim \psi_1$

Transitivität: $\psi_1 \sim \psi_2$ und $\psi_2 \sim \psi_3$: $\psi_1 = \lambda \psi_2$, $\psi_2 = \mu \psi_3 \implies \psi_1 = (\lambda \mu) \psi_3$, $\lambda \mu \in \mathbb{C}$, $\psi_1 \sim \psi_3$

□

Sei $\psi \in \mathcal{H}$. Dann ist die Menge $A = \{\phi \in \mathcal{H} : \phi \sim \psi\}$ eine Äquivalenzklasse bezüglich R .

Satz 2.2 *Jeder Vektor $\psi \in \mathcal{H}$ ist ein Repräsentant genau einer Äquivalenzklasse A bezüglich R .*

Beweis. Angenommen, es gelte für zwei Äquivalenzklassen A_1 und A_2 $A_1 \cap A_2 \neq \emptyset$ und $\psi \in A_1 \cap A_2$.

Sei $\phi \in A_1 \Rightarrow \phi \sim \psi$. Wegen $\psi \in A_2$ folgt $\phi \in A_2$. Also ist $A_1 \subset A_2$. Umgekehrt geht es analog und es folgt $A_1 = A_2$.

□

Damit ist ein wichtiger Schritt getan. In der Modellierung des QC benötigt man ein quantentheoretisches Analogon zum klassischen Bit. Das klassische Bit als kleinste Informationseinheit besitzt zwei mögliche (reine) Zustände: 0 oder 1. Das Analogon muss demnach dieselben Zustände ermöglichen. Die Kopenhagener Deutung erzwingt aber eine wahrscheinlichkeitstheoretische Konstellation, die a priori nicht unterscheiden lässt, ob sich ein Bit im Zustand 0 oder 1 befindet.

Ausgehend von der quantenmechanischen Spinor-Wellenfunktion und der dazu äquivalenten Vektordarstellung entspricht jedem Bit-Zustand dann ein Basisvektor des Hilbertraumes. Die Wahrscheinlichkeitsinterpretation wird mit den längentreuen Linearkombinationen dieser Basisvektoren, die alle wieder zu einem Zustand im Zustandsraum führen, modelliert.

Definition 2.3 (Zustandsraum und q-Bit) *Sei \mathcal{H} ein 2-dimensionaler Hilbertraum über \mathbb{C} . Mit der Festlegung*

$$\mathcal{S}_{\mathcal{H}} \stackrel{\text{def}}{=} \{\psi \in \mathcal{H} : \|\psi\|_{\mathcal{H}}^2 = 1\}$$

*wird die Sphäre $\mathcal{S}_{\mathcal{H}}$ zum **Zustandsraum**. Jedes $\psi \in \mathcal{S}_{\mathcal{H}}$ wird zum **q-Bit** in \mathcal{H} .*

Alle ψ aus dem Zustandsraum $\mathcal{S}_{\mathcal{H}}$ sind Repräsentanten aus einer Äquivalenzklasse A bezüglich der Äquivalenzrelation R aus Satz 2.1 und können durch eine Linearkombination der Basiszustände aus $B = \{b_1, b_2 : b_1, b_2 \in \mathcal{H}\}$ gemäß dem Superpositionsprinzip in Satz 1.1 erzeugt werden.

Satz 2.4 (Superposition von q-Bits) *Jedes ψ aus $\mathcal{S}_{\mathcal{H}}$ erfüllt das Superpositionsprinzip aus Satz 1.1 zusammen mit der Normierungsbedingung gemäß Postulat 1. Umgekehrt liegt jeder Vektor, der mit dem Superpositionsprinzip aus Satz 1.1 zusammen mit der Normierungsbedingung gemäß Postulat 1 gebildet wird, in $\mathcal{S}_{\mathcal{H}}$.*

Beweis. „ \Rightarrow “ Jedes $\psi \in \mathcal{H}$ kann als komplexe Linearkombination aus Basisvektoren dargestellt werden, $\psi = \lambda_1 b_1 + \lambda_2 b_2$. Aus der Normierungsbedingung $\|\psi\|_{\mathcal{H}} = 1$ folgt dann, dass $|\lambda_1|^2 + |\lambda_2|^2 = 1$ gilt.

„ \Leftarrow “ Es ist lediglich zu zeigen, dass für $\psi = \lambda_1 b_1 + \lambda_2 b_2$ dann $\|\psi\|_{\mathcal{H}}^2 = 1$ gilt.

$$\begin{aligned} \|\psi\|_{\mathcal{H}}^2 &= \langle (\lambda_1 b_1 + \lambda_2 b_2) | (\lambda_1 b_1 + \lambda_2 b_2) \rangle_{\mathcal{H}} \\ &= \lambda_1 \bar{\lambda}_1 \langle b_1 | b_1 \rangle + \lambda_1 \bar{\lambda}_2 \langle b_1 | b_2 \rangle + \lambda_2 \bar{\lambda}_1 \langle b_2 | b_1 \rangle + \lambda_2 \bar{\lambda}_2 \langle b_2 | b_2 \rangle \\ &= \lambda_1 \bar{\lambda}_1 + \lambda_2 \bar{\lambda}_2 = 1 \end{aligned}$$

□

ψ lässt sich so interpretieren: Mit einer Wahrscheinlichkeit von $\lambda_1 \bar{\lambda}_1$ wird bei einer Messung der reine Zustand, dem b_1 entspricht, gemessen, mit einer Wahrscheinlichkeit von $\lambda_2 \bar{\lambda}_2$ wird bei einer Messung der reine Zustand, dem b_2 entspricht, gemessen. Das entspricht dem in (1.3) beschriebenen Zufallsexperiment.

Nun soll es aber nicht bei einem q-Bit bleiben. Ein q-Bit besitzt zwei reine Zustände. n q-Bits entsprechen dann 2^n reine Zustände. Das bedeutet, dass für n q-Bits 2^n Basiszustände benötigt werden.

Bemerkung. Der jetzt benötigte dimensionserweiterte Vektorraum $\mathcal{H}^{\otimes n}$ ist wiederum ein Hilbertraum über \mathbb{C} (siehe [Schä02]).

Definition 2.5 (Multi-q-Bit) Sei $\mathcal{H}^{\otimes n}$ ein 2^n -dimensionaler Hilbertraum über \mathbb{C} . Mit der Festlegung

$$\mathcal{S}_{\mathcal{H}^{\otimes n}} \stackrel{\text{def}}{=} \{ \psi \in \mathcal{H}^{\otimes n} : \|\psi\|_{\mathcal{H}^{\otimes n}}^2 = 1 \}$$

wird die Sphäre $\mathcal{S}_{\mathcal{H}^{\otimes n}}$ zum Zustandsraum. Jedes $\psi \in \mathcal{S}_{\mathcal{H}^{\otimes n}}$ ist dann ein **Multi-q-Bit** in $\mathcal{H}^{\otimes n}$.

Bemerkung. Auch hier gilt analog zu Satz 2.4 die Superpositionsangabe. Die Wahrscheinlichkeitsinterpretation kann entsprechend auf das Multi-q-Bit erweitert werden.

Das Multi-q-Bit ψ steht für eine Superposition der Basisvektoren des gegebenen Hilbertraumes $\mathcal{H}^{\otimes n}$. λ_k ist dabei der zum Basisvektor b_k gehörende Linearfaktor. Die anfangs des Kapitels beschriebenen Bit-Zustände können als reine Zustände gemäß Definition 1.2 interpretiert werden. Deshalb soll im Weiteren oBdA. stets die kanonische Basis

$$\{e_1, \dots, e_{2^n}\} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}, \quad e_i \in \mathbb{C}^{2^n}$$

in einem System aus n q-Bits (**n-Bit-System**) betrachtet werden. Ein beliebiger Zustand ψ des Systems wird dann durch

$$\psi = \lambda_1 e_1 + \dots + \lambda_{2^n} e_{2^n} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_{2^n} \end{pmatrix} \in \mathcal{S}_{\mathcal{H}^{\otimes n}} \quad (2.1)$$

beschrieben und zum Multi-q-Bit, wobei die Koeffizienten der Basisvektoren der Normierungsbedingung (1.4) genügen. Jeder Basisvektor e_k , $1 \leq k \leq 2^n$, steht für einen reinen Zustand und so für eine ganz bestimmte Bitfolge, die analog zu [Schä02] von links mit dem ersten Bit beginnend nach rechts bis zum letzten Bit konstruiert wird. Beispiel 2.1.1 zeigt die Repräsentation von Basisvektoren durch Bitkombinationen respektive Bitkombinationen durch Basisvektoren. Zudem wird algorithmisch angegeben, wie in einem n-Bit-System aus den Basisvektoren die Bitkombinationen bzw. aus den Bitkombinationen die Basisvektoren berechnet werden können.

Beispiel 2.1.1

2^n Basisvektoren	assoziierte Zustände aus n Bits
e_1	0000...0
e_2	1000...0
e_3	0100...0
e_4	1100...0
e_5	0010...0
\vdots	\vdots
e_{2^n-1}	0111...1
e_{2^n}	1111...1

Es gibt zwei Richtungen:

1. Gegeben: Bitkombination $B_1 \dots B_n$ aus n Bits. Gesucht: Index k des zugehörigen Basisvektors $e_k \rightarrow$ Algorithmus 1.
2. Gegeben: Basisvektor e_k in n -Bit-System. Gesucht: Bitkombination $B_1 \dots B_n \rightarrow$ Algorithmus 2.

```

1: procedure BESTIMMEBASISVEKTOR( $n, B$ )
2:    $k = 1$ ;
3:   for ( $i = 1, i \leq n$ ) do
4:      $k += B_i \cdot 2^{i-1}$ ;
5:   end for
6:   return;
7: end procedure

```

▷ Basisvektor e_k berechnet.

Algorithmus 1: Bestimmen des Basisvektors e_k aus der Bitkombination $B_1 \dots B_n$ eines n -Bit-Systems.

```

1: procedure ERSTELLEBITKOMBINATION( $n, k$ )
2:    $k \leftarrow -$ ;
3:   while  $n > 0$  do
4:     if  $k < 2^{n-1}$  then
5:        $B_n = 0$ ; ▷ Bit  $n$  nicht gesetzt.
6:     else
7:        $B_n = 1$ ; ▷ Bit  $n$  gesetzt.
8:        $k \leftarrow 2^{n-1}$ ;
9:     end if
10:     $n \leftarrow -$ ;
11:  end while
12:  return; ▷ Bitkombination erstellt.
13: end procedure

```

Algorithmus 2: Bestimmen der Bitkombination aus dem Basisvektor e_k eines n -Bit-Systems.

Messung

Gleichung (2.1) ergibt eine Repräsentation der Wahrscheinlichkeiten für jede Bitfolge durch ein λ_k und das Eintreten eines Zustands kann als Zufallsexperiment modelliert werden. Der Ergebnisraum besteht aus den Basisvektoren: $\Omega = \{e_1, \dots, e_{2^n}\}$. Die Potenzmenge $\mathcal{P}(\Omega)$ besitzt 2^{2^n} Elemente und bildet den Ereignisraum. Die Wahrscheinlichkeit, dass bei einer Messung der zu e_k gehörende reine Zustand ψ_{e_k} angenommen wird, entspricht der Multiplikation des zugehörigen Linearfaktors mit seinem konjugiert-komplexen Wert und induziert so ein Wahrscheinlichkeitsmaß $P : A \rightarrow [0, 1]$ für $A \in \mathcal{P}(\Omega)$ auf $\mathcal{P}(\Omega)$. Denn die Abbildung

$$p : \Omega \rightarrow [0, 1], \quad e_k \mapsto \lambda_k \bar{\lambda}_k$$

ist eine **Zähldichte** (siehe [MeiS05]), da $p(\omega) \geq 0 \quad \forall \omega \in \Omega$ und $\sum_{\omega \in \Omega} p(\omega) = 1$ gilt. Legt man

$P : A \rightarrow [0, 1], P(A) \stackrel{\text{def}}{=} \sum_{\omega \in A} p(\omega)$ fest, so ist nach [MeiS05] P ein Wahrscheinlichkeitsmaß.

Damit lässt sich der Wahrscheinlichkeitsraum $(\Omega, \mathcal{P}(\Omega), P)$ bestimmen und man hat

$$P(\psi_{e_k}) = P(\{e_k\}) = \lambda_k \bar{\lambda}_k = |\lambda_k|^2 \quad (2.2)$$

Der Begriff der Messung ist noch näher zu bestimmen. Das Wahrscheinlichkeitsmaß ist als Zähldichte aufzufassen. Wird nun eine Partitionierung $E_1, \dots, E_l \subseteq \Omega$ mit $E_i \cap E_j = \emptyset, \quad i \neq j$ und $\bigcup_{j=1}^l E_j = \Omega$ in die möglichen Ergebnisse der Messung vorgenommen, so erhält man bei einer Messung das Ergebnis $E_i = \{e_{i_1}, \dots, e_{i_m}\}$ mit der Wahrscheinlichkeit

$$P(E_i) = \sum_{j=1}^m P(\{e_{i_j}\}) = \sum_{j=1}^m |\lambda_{i_j}|^2.$$

Nach der Messung findet die in Abschnitt 1.2 erwähnte Zustandsreduktion statt. Das bedeutet, dass nach der Messung des Ergebnisses E_i das Multi-q-Bit in ein neues Multi-q-Bit $\psi_{neu} =$

$\mu_1 e_1 + \dots + \mu_{2^n} e_{2^n} \in \mathcal{S}_{\mathcal{H}^{\otimes n}}$ übergeht mit

$$\mu_k = \begin{cases} 0 & \text{für } k \notin \{i_1, \dots, i_m\} \\ \frac{\lambda_k}{\sqrt{\sum_{j=1}^m |\lambda_{i_j}|^2}} & \text{für } k \in \{i_1, \dots, i_m\} \end{cases} \cdot \quad (2.3)$$

Eine interessante Eigenschaft von Multi-q-Bits ist die der **Verschränkung**¹ von Zuständen. Ein Zustand ist verschränkt, wenn er nicht als Tensorprodukt einzelner q-Bits darstellbar ist.

Beispiel 2.1.2 Sei in einem 2-Bit-System $\psi \in \mathcal{H}^{\otimes 2}$ das Multi-q-Bit mit folgendem Aussehen:

$$\psi = \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \right] = \frac{1}{\sqrt{2}} (e_3 + e_2) = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

Der Zustand entspricht einer Superposition zweier Basiszustände, die den Bitkombinationen „01“ bzw. „10“ entsprechen.

Bei einer Messung des ersten Bits in dem Beispiel, d.h. es wird eine Partitionierung in $E_1 = \{e_1, e_3\}$ und $E_2 = \{e_2, e_4\}$ vorgenommen, wird dann mit Wahrscheinlichkeit $P(E_1) = P(\{e_1\}) + P(\{e_3\}) = \frac{1}{2}$ und $P(E_2) = P(\{e_2\}) + P(\{e_4\}) = \frac{1}{2}$ der Wert 0 oder 1 gemessen. Ist das jedoch geschehen, ergibt sich eine Verschränkung von Zuständen: Wurde 0 gemessen, so geht das System in den Zustand $\psi_{neu} = e_3$ über und das zweite Bit ist sicher auf 1, wurde andererseits 1 gemessen, geht das System in den Zustand $\psi_{neu} = e_2$ über und das zweite Bit ist sicher auf 0. Der gemessene Wert des ersten Bits ist völlig zufällig, der des zweiten Bits dann aber festgelegt. Das analoge Verhalten zeigt sich, wenn zunächst das zweite Bit gemessen wird.

An diesem Punkt endet die Betrachtung des statischen Systems. Die zeitliche Entwicklung des Systems und damit das dritte Postulat der Quantenmechanik soll jetzt untersucht werden.

2.1.2 Modellierung der zeitlichen Dynamik

Gemäß Postulat 3 ist der Zeitentwicklungsoperator eines konservativen Systems unitär. Die dritte Implikation in (1.12) findet nun sofortige Anwendung. Eine unitäre Matrix $M \in \mathcal{H}^{\otimes n, \otimes n}$, bildet ein Multi-q-Bit $\psi \in \mathcal{H}^{\otimes n}$ auf ein neues Multi-q-Bit ϕ ab.

$$\psi \text{ Multi-q-Bit} \Rightarrow \phi = M \cdot \psi \text{ ist Multi-q-Bit} \quad (2.4)$$

Das führt auf den Begriff einer Operation auf der Menge der Multi-q-Bits. Denn die Menge der unitären Matrizen in $\mathcal{H}^{\otimes n, \otimes n}$ ist eine multiplikative Gruppe mit der Einheitsmatrix als Einselement und der inversen Matrix als inversem Element.

¹Der Begriff Verschränkung stammt von Schrödinger (12.08.1887-04.01.1961) aus dem Jahre 1935. Der heute allgemein übliche Terminus lautet „Entanglement“ [Niel00]

Definition 2.6 (Operation auf Multi-q-Bits, Gate) Sei $\mathcal{S}_{\mathcal{H}^{\otimes n}}$ ein Zustandsraum. M sei linearer unitärer Operator auf $\mathcal{H}^{\otimes n}$ und $\psi \in \mathcal{S}_{\mathcal{H}^{\otimes n}}$. Die Wirkung von M auf ψ , die gemäß (2.4) das Multi-q-Bit auf ein neues Multi-q-Bit abbildet,

$$\begin{aligned}\mathcal{H}^{\otimes n} &\rightarrow \mathcal{H}^{\otimes n} \\ \psi &\mapsto M \cdot \psi\end{aligned}$$

heißt **Operation auf einem Multi-q-Bit**. Die Matrix M wird als **Gate** bezeichnet.

Bemerkung. Operationen auf Multi-q-Bits sind reversibel. Eine Messung dagegen ist nicht mehr umkehrbar (vgl. [Chil00]).

2.2 Konstruktion von Gates

Das Ziel dieses Abschnitts ist es, zu einem gegebenen Gate eine Zerlegung in „einfachere“ Gates zu finden, die jeweils Operationen auf einem einzigen q-Bit durchführen. Die Anwendung eines solchen speziellen Gates hat den Rechenaufwand einer Operation in einem Quantencomputer. Für eine klassische Berechnung benötigt jedes derartige Gate q-Bit-unabhängig stets gleich viele Rechenoperationen. Klassischer und quantentheoretischer Rechenaufwand sind damit vergleichbar. Hierzu muss zunächst geklärt werden, was unter einer Operation auf einem einzelnen q-Bit zu verstehen ist.

2.2.1 Operationen auf einzelnen Bits

Sei nun ein n-Bit-System mit dem Zustandsraum $\mathcal{S}_{\mathcal{H}^{\otimes n}}$ und der Zustandsdarstellung wie in (2.1) gegeben. Im Folgenden werden, wie im Beispiel 2.2.1 vorgenommen, die reinen Zustände eines n-Bit-Systems jeweils in zwei Mengen partitioniert.

Bitnummer m	Bit m=0	Bit m=1
1	000 e_1 ψ_{000}	100 e_2 ψ_{100}
	010 e_3 ψ_{010}	110 e_4 ψ_{110}
	001 e_5 ψ_{001}	101 e_6 ψ_{101}
	011 e_7 ψ_{011}	111 e_8 ψ_{111}
2	000 e_1 ψ_{000}	010 e_3 ψ_{000}
	100 e_2 ψ_{100}	110 e_4 ψ_{100}
	001 e_5 ψ_{001}	011 e_7 ψ_{001}
	101 e_6 ψ_{101}	111 e_8 ψ_{111}
3	000 e_1 ψ_{000}	001 e_5 ψ_{001}
	100 e_2 ψ_{100}	101 e_6 ψ_{101}
	010 e_3 ψ_{010}	011 e_7 ψ_{011}
	110 e_4 ψ_{110}	111 e_8 ψ_{111}

Tafel 2.1: Partitionierung der Zustände nach dem m -ten Bit

Beispiel 2.2.1 Betrachtet wird das 3-Bit-System in Tafel 2.1. Es wird zunächst das Bit m ausgewählt, nach dem die Zustände partitioniert werden sollen. Danach werden zwei Mengen erstellt: Die Eine enthält alle möglichen Zustände des Systems, bei denen das m -te Bit nicht gesetzt ist, die Zweite enthält entsprechend alle möglichen Zustände, deren m -tes Bit gesetzt ist. Somit sind beide Mengen disjunkt.

Auf diese Weise kann gemäß den Beispielen 2.1.2 und 2.2.1 allgemein eine Partitionierung abhängig von der Wahl eines bestimmten Bits definiert werden.

Definition 2.7 (Bit- m -Partitionierung) Wird in einem n -Bit-System eine Aufteilung der Menge Ω aller reinen Zustände in zwei disjunkte Mengen E_{m0} und E_{m1} erstellt (d.h. $\Omega = E_{m0} \cup E_{m1}$ und $E_{m0} \cap E_{m1} = \emptyset$), wobei E_{m0} alle Zustände, bei denen das m -te Bit des Systems nicht gesetzt ist, und E_{m1} die Zustände, bei denen das m -te Bit des Systems gesetzt ist, enthält,

$$E_{m0} \stackrel{\text{def}}{=} \{\psi_{e_k} \in \Omega : \text{Bit } m \text{ von } \psi_{e_k} \text{ ist nicht gesetzt}\}$$

$$E_{m1} \stackrel{\text{def}}{=} \{\psi_{e_k} \in \Omega : \text{Bit } m \text{ von } \psi_{e_k} \text{ ist gesetzt}\}$$

ergibt sich eine **Bit- m -Partitionierung**.

Satz 2.8 Beide Mengen einer Bit- m -Partitionierung eines n -Bit-Systems enthalten jeweils genau 2^{n-1} Elemente.

Beweis. Gegeben sei eine Bit- m -Partitionierung eines n -Bit-Systems mit $\Omega = E_{m0} \cup E_{m1}$ und $E_{m0} \cap E_{m1} = \emptyset$. Ein beliebiger Zustand $\psi_{e_k} \in E_{m0}$ hat mit $B_i \in \{0, 1\}$, $i = 1, \dots, n$, $i \neq m$ die Gestalt

$$B_1 \dots B_{m-1} \ 0 \ B_{m+1} \dots B_n \quad B_m = 0 \tag{2.5}$$

und zu jedem dieser ψ_{e_k} existiert ein $\psi_{e_l} \in \Omega$ der Gestalt

$$B_1 \dots B_{m-1} \ 1 \ B_{m+1} \dots B_n \quad B_m = 1 \tag{2.6}$$

mit $\psi_{e_l} \in E_{m1}$. $\Rightarrow E_{m1}$ hat mindestens genauso viele Elemente wie E_{m0}

Analog gibt es umgekehrt zu jedem Zustand $\psi_{e_l} \in E_{m1}$ einen Zustand $\psi_{e_k} \in E_{m0}$, so dass E_{m0} mindestens genauso viele Elemente wie E_{m1} hat.

\Rightarrow Beide Mengen sind gleichmächtig und die Anzahl der Elemente teilt sich gleichmäßig auf beide Mengen zu je 2^{n-1} Zuständen auf.

□

Bemerkung. Die in dem Beweis benutzte **Korrespondenz** zweier Zustände wird bei der Zerlegung von Gates in Abschnitt 2.3 eine wesentliche Rolle spielen.

Mit einer Bit- m -Partitionierung können nun zu einem gegebenen Multi- q -Bit die Wahrscheinlichkeiten dafür berechnet werden, dass bei einer Messung das m -te Bit gesetzt oder nicht gesetzt sein wird. Dazu wird der Wahrscheinlichkeitsraum $(\Omega, \{\emptyset, \Omega, E_{m0}, E_{m1}\}, P)$ betrachtet und es werden die einzelnen Zustandswahrscheinlichkeiten gemäß Gleichung (2.2) in jeder der beiden Mengen zu einer Gesamtwahrscheinlichkeit für jede Menge addiert.

$$\begin{aligned} P(E_{m0}) &= \sum_{\substack{k=1 \\ \psi_{e_k} \in E_{m0}}}^{2^n} P(\psi_{e_k}) \\ P(E_{m1}) &= \sum_{\substack{k=1 \\ \psi_{e_k} \in E_{m1}}}^{2^n} P(\psi_{e_k}) \end{aligned} \quad (2.7)$$

Beispiel 2.2.2 Betrachtet wird die Partitionierung für Bit 2 aus Beispiel 2.2.1. Zu dem Multi- q -Bit

$$\psi = \frac{1}{8} \begin{pmatrix} 3 & -2 & -3 & 4 & i & 0 & 5 & 0 \end{pmatrix}^T$$

ergeben sich folgende Wahrscheinlichkeiten für den Zustand für Bit 2:

$$\begin{aligned} P(\psi_{20}) &= P(\psi_{e_1}) + P(\psi_{e_2}) + P(\psi_{e_5}) + P(\psi_{e_6}) \\ &= \frac{1}{64} (3 \cdot 3 + (-2) \cdot (-2) + i \cdot (-i) + 0 \cdot 0) = \frac{14}{64} = \frac{7}{32} \\ P(\psi_{21}) &= P(\psi_{e_3}) + P(\psi_{e_4}) + P(\psi_{e_7}) + P(\psi_{e_8}) \\ &= \frac{1}{64} ((-3) \cdot (-3) + 4 \cdot 4 + 5 \cdot 5 + 0 \cdot 0) = \frac{50}{64} = \frac{25}{32} \end{aligned}$$

2.2.2 Bit- m -Operationen

Eine beliebige Operation auf einem Multi- q -Bit kann die Wahrscheinlichkeiten für ein q -Bit verändern. Ziel ist nun, solche Operationen zu finden, die ausschließlich imstande sind, Wahrscheinlichkeiten für ein festgelegtes q -Bit zu ändern.

Definition 2.9 (Bit- m -Operation) Eine Operation auf einem beliebigen Multi- q -Bit, die Zustandswahrscheinlichkeiten für das m -te Bit verändern kann und die Zustandswahrscheinlichkeiten für alle anderen Bits unverändert lässt, heißt **Bit- m -Operation**.

Eine Bit- m -Operation stellt besondere Anforderungen an ein Gate, die zu bestimmen sind. Nicht jedes Gate vermag nur eine einzige Bit- m -Operation durchzuführen. Betrachtet wird eine unitäre Matrix $M = (m_{ij}) \in \mathcal{H}^{\otimes n, \otimes n}$ angewendet auf ein Multi- q -Bit $\psi \in \mathcal{H}^{\otimes n}$, $\psi = \sum_{i=1}^{2^n} \lambda_i e_i$

und $\psi_{neu} = \sum_{i=1}^{2^n} \mu_i e_i \stackrel{def}{=} M\psi$, so bestimmt die k -te Zeile in M zusammen mit ψ den neuen Gewichtungsfaktor μ_k von e_k .

$$\mu_k = \sum_{i=1}^{2^n} m_{ki} \lambda_i.$$

Es ergibt sich eine neue Zustandswahrscheinlichkeit $P(\psi_{neu, e_k})$. Ein Eintrag in der k -ten Spalte mit $m_{kk} \bar{m}_{kk} < 1$ verringert $P(\psi_{e_k})$. Nun gilt aber gemäß der Unitäreigenschaft (1.11)

$$\sum_{i=1}^{2^n} m_{ik} \bar{m}_{ik} = 1 \quad (2.8)$$

In wenigstens einer Zeile $l \neq k$ in M muss es dann einen Eintrag $m_{lk} \neq 0$ geben. Damit M eine Bit- m -Operation darstellen kann, darf nicht in jeder Zeile $l \neq k$ ein Eintrag stehen. Denn mit einem Eintrag in m_{lk} wird $P(\psi_{e_l})$ verändert.

Satz 2.10 (Konstruktion einer Bit- m -Operation) *Ein Gate M ist genau dann eine Bit- m -Operation, wenn in jeder Spalte k , $1 \leq k \leq 2^n$, von 0 verschiedene Einträge höchstens bei m_{kk} und bei dem bezüglich m korrespondierenden m_{lk} mit*

$$l = \begin{cases} k + 2^{m-1} & \text{für Bit } B_m = 0 \\ k - 2^{m-1} & \text{für Bit } B_m = 1 \end{cases} \quad (2.9)$$

vorkommen.

Beweis. Zu beweisen sind zwei Richtungen. „ \Rightarrow “ Sei das Gate M eine Bit- m -Operation, das auf den reinen Zustand $\psi_{e_k} = e_k$ angewendet werden soll. Dann wird e_k auf die Spalte m_k abgebildet. Steht nun ψ_{e_k} oBdA. für das Bit wie in (2.5), so gibt es genau einen Zustand ψ_{e_l} , der die Bitkombination wie in (2.6) darstellt. Jeder andere Zustand ψ_{e_j} , $j \neq k$ und $j \neq l$ unterscheidet sich in mindestens einem anderen Bit als Bit m von ψ_{e_k} und ψ_{e_l} . Das bedeutet, dass es eine Änderung der Wahrscheinlichkeitsverteilung für das zu ψ_{e_j} gehörige alternative Bit gibt. Die Wahrscheinlichkeit, dass dieses Bit den komplementären Wert annimmt, wird zu einem Wert $|m_{jk}|^2 > 0$ erhöht. Das kann aber nicht mehr ausgeglichen werden. Es dürfen damit nur Spalteneinträge in m_{kk} und m_{lk} stehen, um die Bit- m -Operation nicht zu zerstören. Da die Werte der Bits bis auf B_m gleich sind, ist die Differenz zwischen k und l genau 2^{m-1} . Steht Bit m auf dem Wert 0 wie in (2.5), so ergibt sich l durch $l = k + 2^{m-1}$. Steht Bit m auf dem Wert 1 wie in (2.6), dann ergibt sich l durch $l = k - 2^{m-1}$. Den Wert B_m von ψ_{e_k} erhält man durch Betrachten der Bitkombination. Durch „Abschneiden“ der Bits B_{m+1} bis B_n durch den Modulo-Operator und einen Bit-Shift nach links um $m - 1$ mit dem Division-ohne-Rest-Operator bleibt der gesuchte Bitwert B_m übrig (die Korrektur von k um -1 entsteht durch die Identifizierung des ersten Basisvektors mit der Bitfolge aus lauter Nullen, vgl. Algorithmen 1 bzw. 2):

$$B_m = ((k - 1) \bmod 2^m) \operatorname{div} 2^{m-1}$$

„ \Leftarrow “ Besitze das Gate M in jeder Spalte k höchstens bei m_{kk} oder m_{lk} mit l wie in (2.9) vorgesehen Einträge. Bei Anwendung auf ein beliebiges Multi-q-Bit ψ kann eine spaltenweise Betrachtung vorgenommen werden, da sich ψ als Linearkombination der Basisvektoren gemäß (2.1) darstellen lässt. Die einzelnen Spalten von M werden skaliert und zu einem Gesamtvektor aufaddiert: $\psi_{neu} = M\psi = \sum_{k=1}^{2^n} \sum_{i=1}^{2^n} m_{ki} \lambda_i e_k$. Jedes λ_k wird gespalten in Anteile für e_k und das bezüglich Bit m korrespondierende e_l , analog für λ_l und es ergeben sich die vier Summanden:

$$\underbrace{m_{kk}\lambda_k e_k + m_{kl}\lambda_l e_k}_{\mu_k e_k} + \underbrace{m_{ll}\lambda_l e_l + m_{lk}\lambda_k e_l}_{\mu_l e_l}.$$

Also werden insgesamt Anteile nur zwischen dem m -ten Bit ausgetauscht und alle anderen Bits bleiben dann in Bezug auf die Wahrscheinlichkeitswerte unverändert.

□

Beispiel 2.2.3 Für $n = 3$ werden nun Beispiele für Matrizen von Bit-1-, Bit-2- und Bit-3-Operationen dargestellt.

Bit-1-Operation:

$$M_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{2} & \frac{\sqrt{3}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & i & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{10}} & \frac{-3}{\sqrt{10}} \end{pmatrix}$$

Bit-2-Operation:

$$M_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{i}{\sqrt{10}} & 0 & \frac{3}{\sqrt{10}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & \frac{3}{\sqrt{10}} & 0 & \frac{i}{\sqrt{10}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} \end{pmatrix}$$

Bit-3-Operation:

$$M_3 = \begin{pmatrix} \frac{3}{\sqrt{5}} & 0 & 0 & 0 & \frac{3}{\sqrt{5}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{3}{\sqrt{5}} & 0 & 0 & 0 & \frac{-4}{\sqrt{5}} \\ \frac{4}{\sqrt{5}} & 0 & 0 & 0 & \frac{-3}{\sqrt{5}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & \frac{4}{\sqrt{5}} & 0 & 0 & 0 & \frac{3}{\sqrt{5}} \end{pmatrix}$$

Bemerkung. Die im letzten Beispiel dargestellten Bit-m-Operationen können als jeweils vier (multiplikativ) hintereinander ausgeführte Bit-m-Operationen interpretiert werden, die für sich genommen Rotations- oder Spiegelungsmatrizen sind. Ein allgemein bekannter Vertreter solcher Rotationsmatrizen ist die Givens-Rotation (vgl. [Deuf93]).

Es gibt Gates, die verschiedene Einträge haben und bei der Anwendung auf Multi-q-Bits die Zustandswahrscheinlichkeiten in gleicher Weise verändern.

Definition 2.11 (Wirkungsgleichheit von Gates) Zwei verschiedene Gates $M, N \in \mathcal{H}^{\otimes n, \otimes n}$ heißen *wirkungsgleich*, falls bei der Anwendung auf ein beliebiges Multi-q-Bit $\psi \in \mathcal{S}_{\mathcal{H}^{\otimes n}}$ gilt

$$P_{M\psi}(E_{m0}) = P_{N\psi}(E_{m0}) \quad \text{und} \quad P_{M\psi}(E_{m1}) = P_{N\psi}(E_{m1}) \quad \forall m = 1, \dots, n$$

In diesem Abschnitt wurde gezeigt, wie Gates durch die multiplikative Kombination einzelner Bit-m-Operationen konstruiert werden können. Das wird für das später in Kapitel 4 vorgestellte Branch&Bound-Verfahren von Bedeutung sein. Diesbezüglich genauso wichtig ist der folgende Abschnitt. Er behandelt den umgekehrten Weg, wie ein gegebenes Gate in einzelne Bit-m-Operationen zerlegt werden kann.

2.3 Zerlegung von Gates

Geht man von einer gegebenen unitären Matrix aus, die ein Gate repräsentiert, so stellt sich die Frage, ob dieses Gate in Operationen zerlegt werden kann, die jeweils nur die Wahrscheinlichkeitsverteilung eines Bits verändern, d.h. die eine Bit-m-Operation darstellen.

2.3.1 Zerlegbarkeit von Bit-m-Operationen

Eine unitäre Matrix lässt sich als Produkt anderer unitärer Matrizen darstellen. Sind alle einzelnen Faktoren Bit-m-Operationen, so lässt sich die unitäre Matrix in Bit-m-Operationen zerlegen.

Definition 2.12 (Zerlegbarkeit) *Ein Gate in Form einer unitären Matrix $M \in \mathcal{H}^{\otimes n, \otimes n}$ heißt zerlegbar in einzelne Bit-m-Operationen, falls es Bit-m-Operationen $R_1, \dots, R_k \in \mathcal{H}^{\otimes n, \otimes n}$ gibt mit*

$$R_1 \cdot \dots \cdot R_k = M \quad (2.10)$$

Beispiel 2.3.1 *An einem einfachen Beispiel kann gezeigt werden, dass die Zerlegung eines Gates in Bit-m-Operationen nicht eindeutig ist. Sowohl die Einträge der Matrizen der jeweiligen Bit-m-Operationen und damit ihre Einzelwirkung auf ein Multi-q-Bit, als auch die Reihenfolge ihrer Anwendung auf ein Multi-q-Bit (vgl. (1.13)) ist nicht eindeutig bestimmbar.*

$$\frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$\frac{1}{\sqrt{2}} \begin{pmatrix} i & 0 & -i & 0 \\ 0 & 1 & 0 & 1 \\ i & 0 & i & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} -i & -i & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & i & i \\ 0 & 0 & 1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

Es ist zu zeigen, wie eine Zerlegung einer unitären Matrix $M \in \mathcal{H}^{\otimes n, \otimes n}$ durch andere unitäre Matrizen bestimmt werden kann. Allgemein lassen sich normale Matrizen $A \in \mathcal{H}^{k, k}$, $A^\dagger A = AA^\dagger$, mit Hilfe unitärer Matrizen $U \in \mathcal{H}^{k, k}$ mit

$$U^\dagger A U = \text{diag}(\lambda_1, \dots, \lambda_k) \quad , \quad \lambda_i, i = 1, \dots, k \quad \text{Eigenwerte von } A$$

diagonalisieren (vgl. [Råde97]). Da aber M unitär ist, genügt die Multiplikation mit der zu M inversen Matrix M^\dagger , um M zur Einheitsmatrix zu diagonalisieren. Um eine Verbindung zwischen M^\dagger und Bit-m-Operationen herzustellen, kann zunächst überlegt werden, dass mit Hilfe von unitären Rotations- oder Spiegelungsmatrizen - hier zusammen als Transformationsmatrizen bezeichnet - die Matrix M^\dagger konstruiert werden kann. Das zeigt der folgende Satz.

jede Transformationsmatrix R_j sind unitär und deshalb ist auch das Produkt P dieser Matrizen gemäß (1.13) unitär. So müssen gemäß (1.11) die Spalten normiert und paarweise orthogonal sein. Da jedoch in der ersten Spalte nur das Diagonalelement von 0 verschieden ist, müssen alle weiteren Elemente von P in der ersten Zeile 0 sein. Dies wird sukzessive für alle Zeilen fortgeführt und belegt so die Diagonalgestalt von P . Für die Diagonalelemente gilt dann $|p_{ii}| = 1 \quad \forall i = 1, \dots, 2^n$. Wird die Diagonalmatrix P noch auf die andere Seite gebracht, erhält man die Matrix M^\dagger .

□

Bemerkung. Die Diagonalisierung einer unitären Matrix kann analog auch durch Rechts-Transformationen oder durch eine Mischung aus beiden Transformationsmöglichkeiten durchgeführt werden.

Bei der eben durchgeführten Diagonalisierung werden Transformationsmatrizen R_j verwendet, die keine Bit-m-Operationen sind. Um aber eine Zerlegung einer unitären Matrix in Bit-m-Operationen zu erreichen, müssen alle Transformationsmatrizen R_j in Bit-m-Operationen zerlegbar sein.

Satz 2.14 (Zerlegbarkeit von Transformationsmatrizen) *Jede Transformationsmatrix R_j kann in höchstens $2n - 1$ Bit-m-Operationen zerlegt werden.*

Beweis. Eine gegebene Transformationsmatrix R_j ist unitär. Seien k, l die Zeilen in R_j , bei denen die Transformation durchgeführt wird. Die Idee besteht nun darin, eine Korrespondenzkette aufzubauen, die am einen Ende e_k und am anderen Ende e_l hat. Werden die zu e_k und e_l gehörenden Bitdarstellungen betrachtet, kann die Kette aufgebaut werden. Sei e_k der Ausgangspunkt der Kette. Unterscheidet sich das erste Bit in beiden Darstellungen, so muss eine Bit-1-Operation bezüglich e_k und dem dazu korrespondierenden Element e_{k_1} in Form einer Vertauschung V_1 durchgeführt werden, um die Bit-Darstellungen anzugleichen. Unterscheidet sich das zweite Bit, muss eine Bit-2-Operation bezüglich e_{k_1} und dem dazu korrespondierenden Element e_{k_2} wiederum mit einer Vertauschung V_2 durchgeführt werden. Sukzessive wird das mit den weiteren sich unterscheidenden Bits, bis zum Vorletzten, durchgeführt. Nach i ($i < n$) Schritten wird beim letzten sich unterscheidenden Bit eine Bit-m-Operation M_{i+1} zwischen e_{k_i} und e_l durchgeführt. Die Operationsmatrix besteht aus den von Null verschiedenen Elementen $m_{k_i k_i}^{i+1} = \bar{r}_{kk}^j$, $m_{l k_i}^{i+1} = \bar{r}_{kl}^j$, $m_{k_i l}^{i+1} = \bar{r}_{lk}^j$ und $m_{ll}^{i+1} = \bar{r}_{ll}^j$ und sonst Einsen auf der Diagonalen. Die Vertauschungen werden nun in umgekehrter Reihenfolge noch einmal durchgeführt. Es werden insgesamt höchstens $n - 1 + 1 + n - 1 = 2n - 1$ Bit-m-Operationen benötigt und die Einheitsmatrix $\mathbb{1}_{2^n} = V_1 \cdot \dots \cdot V_i \cdot M_{i+1} \cdot V_i \cdot \dots \cdot V_1 \cdot R_j$ ist entstanden. Durch Umformung erhält man eine Darstellung von R durch Bit-m-Operationen, $R_j = V_1 \cdot \dots \cdot V_i \cdot M_{i+1}^\dagger \cdot V_i \cdot \dots \cdot V_1$.

□

Beispiel 2.3.2 Zerlegung einer Transformationsmatrix:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$k = 1, l = 4$, Bitdarstellungen: 00 und 11. Erstes Bit unterscheidet sich, deswegen Bit-1-Operation (Vertauschung V) betreffend e_1 und e_2 . Dann M_2 und zuletzt wieder die Vertauschung V durchführen: $R_j = V \cdot M_2^\dagger \cdot V$.

Nun folgt der entscheidende Satz für die Zerlegung von Gates in einzelne Bit-m-Operationen.

Satz 2.15 (Zerlegung von Gates in Bit-m-Operationen) Jedes in Form einer unitären Matrix $M \in \mathcal{H}^{\otimes n, \otimes n}$ repräsentierte Gate lässt sich gemäß Definition 2.12 in Bit-m-Operationen $R_1, \dots, R_k \in \mathcal{H}^{\otimes n, \otimes n}$ zerlegen mit

$$R_k \cdot \dots \cdot R_1 M = \text{diag}(\lambda_1 \dots \lambda_{2^n}) \quad , \quad |\lambda_i| = 1 \quad \forall i = 1, \dots, 2^n \quad (2.12)$$

$$M = R_1^\dagger \cdot \dots \cdot R_k^\dagger \cdot \text{diag}(\lambda_1, \dots, \lambda_{2^n}) \quad (2.13)$$

Beweis. Mit dem Satz über die Diagonalisierbarkeit unitärer Matrizen gibt es eine Diagonalisierung für ein gegebenes Gate mit Hilfe von Transformationsmatrizen. Da Transformationsmatrizen nach dem vorhergehenden Satz in Bit-m-Operationen zerlegbar sind, folgt unmittelbar die Gleichung (2.12). Alle diese Matrizen sind unitär und da $\text{diag}(\lambda_1 \dots \lambda_{2^n})$ ebenfalls eine Bit-m-Operation ist, lässt sich die Gleichung (2.13) direkt ableiten.

□

Beispiel 2.3.3 Das Gate

$$\frac{1}{8} \begin{pmatrix} -6 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & -6 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & -6 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & -6 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & -6 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & -6 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & -6 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & -6 \end{pmatrix}$$

kann in Bit-m-Operationen zerlegt werden. In Tafel 2.2 sind die ersten sieben erforderlichen Transformationsmatrizen aufgelistet, um in der ersten Spalte wie im Beweis zu Satz 2.13 benutzt unterhalb der Diagonalen Nullen zu erzeugen. Spalte zwei in Tafel 2.2 zeigt, auf welchen Zeilen die jeweilige Matrix wirkt, Spalte drei zeigt die vier nichttrivialen Elemente der Matrix. Falls es sich bei der Transformation um keine Bit-m-Operation handelt (zu sehen in Spalte vier), werden gemäß Spalte fünf dementsprechend viele Zerlegungsmatrizen benötigt, um Bit-m-Operationen zu erhalten.

Matrix Nr.	Zeilen	Block	Bit-m-Operation	Anz. Zerlegungsop.
1	1, 8	$\begin{pmatrix} -\frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ -\frac{1}{\sqrt{10}} & -\frac{3}{\sqrt{10}} \end{pmatrix}$	—	5
2	1, 7	$\begin{pmatrix} \sqrt{\frac{10}{11}} & \frac{1}{\sqrt{11}} \\ -\frac{1}{\sqrt{11}} & \sqrt{\frac{10}{11}} \end{pmatrix}$	—	3
3	1, 6	$\begin{pmatrix} \frac{1}{2}\sqrt{\frac{11}{3}} & \frac{1}{2\sqrt{3}} \\ -\frac{1}{2\sqrt{3}} & \frac{1}{2}\sqrt{\frac{11}{3}} \end{pmatrix}$	—	3
4	1, 5	$\begin{pmatrix} 2\sqrt{\frac{3}{13}} & \frac{1}{\sqrt{13}} \\ -\frac{1}{\sqrt{13}} & 2\sqrt{\frac{3}{13}} \end{pmatrix}$	(3)✓	—
5	1, 4	$\begin{pmatrix} \sqrt{\frac{13}{14}} & \frac{1}{\sqrt{14}} \\ -\frac{1}{\sqrt{14}} & \sqrt{\frac{13}{14}} \end{pmatrix}$	—	3
6	1, 3	$\begin{pmatrix} \sqrt{\frac{14}{15}} & \frac{1}{\sqrt{15}} \\ -\frac{1}{\sqrt{15}} & \sqrt{\frac{14}{15}} \end{pmatrix}$	(2)✓	—
7	1, 2	$\begin{pmatrix} \frac{\sqrt{15}}{4} & \frac{1}{4} \\ -\frac{1}{4} & \frac{\sqrt{15}}{4} \end{pmatrix}$	(1)✓	—

Tafel 2.2: Die ersten 7 Transformationen zur Zerlegung der Beispielmatrix.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\sqrt{\frac{3}{5}} & \frac{1}{\sqrt{15}} & \frac{1}{\sqrt{15}} & \frac{1}{\sqrt{15}} & \frac{1}{\sqrt{15}} & \frac{1}{\sqrt{15}} & \frac{1}{\sqrt{15}} \\ 0 & \sqrt{\frac{3}{70}} & -\frac{11}{\sqrt{210}} & 2\sqrt{\frac{2}{105}} & 2\sqrt{\frac{2}{105}} & 2\sqrt{\frac{2}{105}} & 2\sqrt{\frac{2}{105}} & 2\sqrt{\frac{2}{105}} \\ 0 & \frac{3}{\sqrt{182}} & \frac{3}{\sqrt{182}} & -5\sqrt{\frac{2}{91}} & 2\sqrt{\frac{2}{91}} & 2\sqrt{\frac{2}{91}} & 2\sqrt{\frac{2}{91}} & 2\sqrt{\frac{2}{91}} \\ 0 & \frac{1}{2}\sqrt{\frac{3}{13}} & \frac{1}{2}\sqrt{\frac{3}{13}} & \frac{1}{2}\sqrt{\frac{3}{13}} & -\frac{3}{2}\sqrt{\frac{3}{13}} & \frac{2}{\sqrt{39}} & \frac{2}{\sqrt{39}} & \frac{2}{\sqrt{39}} \\ 0 & \frac{1}{2}\sqrt{\frac{3}{11}} & \frac{1}{2}\sqrt{\frac{3}{11}} & \frac{1}{2}\sqrt{\frac{3}{11}} & \frac{1}{2}\sqrt{\frac{3}{11}} & -\frac{4}{\sqrt{33}} & \frac{2}{\sqrt{33}} & \frac{2}{\sqrt{33}} \\ 0 & \frac{3}{\sqrt{110}} & \frac{3}{\sqrt{110}} & \frac{3}{\sqrt{110}} & \frac{3}{\sqrt{110}} & \frac{3}{\sqrt{110}} & -\frac{7}{\sqrt{110}} & 2\sqrt{\frac{2}{55}} \\ 0 & -\frac{1}{\sqrt{10}} & -\frac{1}{\sqrt{10}} & -\frac{1}{\sqrt{10}} & -\frac{1}{\sqrt{10}} & -\frac{1}{\sqrt{10}} & -\frac{1}{\sqrt{10}} & \sqrt{\frac{2}{5}} \end{pmatrix}$$

Die Matrix zeigt das Ergebnis nach Anwendung der Transformationen. In Tafel 2.3 ist umgekehrt zu sehen, wie die Beispielmatrix konstruiert wird. Dabei genügen fünf Bit-m-Operationen.

Das in Kapitel 4 vorgestellte Verfahren zielt darauf ab, einen gegebenen QA, der aus einer Abfolge von Bit-m-Operationen besteht, die auf ein Start-Multi-q-Bit angewendet werden, effizient berechnen zu können. Das setzt bei der Aufstellung der Gates schon Effizienz voraus. Sowohl bei der Konstruktion von Gates als auch bei der Zerlegung von Gates muss auf zwei Regeln geachtet werden:

- Aufeinanderfolgende Operationen auf dem selben Bit werden wenn möglich zu einer Operation zusammengefasst.

Name	Bit	Transformationsmatrix
R_1	1	$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$
R_2	2	$\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix}$
R_3	3	$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$
R_4	2	$\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix}$
R_5	1	$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$

Tafel 2.3: Bit-m-Operationen und deren Operationsbits der Matrix aus Beispiel 2.3.3 der Reihenfolge nach geordnet, $M = R_1 \cdot R_2 \cdot R_3 \cdot R_4 \cdot R_5$.

- Wird bei einer Zerlegung in einer Spalte begonnen, Nullen zu erzeugen, so soll zunächst diese Spalte komplett mit Nullen belegt werden.

Bei der ersten Regel ist der Ausdruck „wenn möglich“ von Bedeutung. Es zeigt sich, dass es nicht immer möglich ist, Operationen auf den selben Bits zusammenzufassen. Beispiele dafür sind Black-Box-basierende QA (vgl. 3.3.2), bei denen die Black-Box oftmals aus einer Einheitsmatrix mit einem oder wenigen Einträgen „-1“ besteht. Eine solche Matrix wäre theoretisch mit jeder Bit-m-Operation zusammenfassbar, aus der algorithmischen Konstruktion heraus darf das

aber nicht getan werden. Die zweite Regel gilt für die Zerlegung eines gegebenen Gates. Dabei ist eine systematische spaltenweise Vorgehensweise geboten. Zunächst können in beliebigen Spalten Nullen erzeugt werden. Nachfolgende Transformationen jedoch zerstören nicht systematisch erzeugte Nullen wieder. Das kann zu einer unnötig hohen Zahl an Bit-m-Operationen führen, was für die Effizienz einer Simulation des QA nicht dienlich wäre.

Bei einer Zerlegung stehen demnach die Systematik und mögliche Einsparungen im Zentrum und das soll bei dem nun folgenden Abschnitt für einen Zerlegungsalgorithmus ebenfalls benutzt werden.

2.3.2 Zerlegungsalgorithmus

Bei der Zerlegung eines gegebenen Gates in einem n -Bit-System kann zunächst der allgemeine Ansatz gemacht werden, unterhalb der Diagonalen lauter Nullen zu erzeugen. Das führt dann automatisch zur Diagonalisierung des Gates und benötigt somit Bit-m-Operationen in der Größenordnung

$$\mathcal{O}(2n \cdot (2^n - 1) \cdot 2^{n-1})$$

(vgl. Sätze 2.14 bzw. 2.15 und [Born03]). Mit Hilfe einer Entwicklung über die Dimension der Gates, d.h. über die Anzahl der q -Bits, lässt sich ein Zerlegungsalgorithmus darstellen. Ist x_n die maximale Anzahl der benötigten Bit-m-Operationen in einem n -Bit-System, so lässt sich zeigen, dass im $(n+1)$ -Bit-System zusätzlich

$$x_n + 2^n - n$$

Operationen gebraucht werden und dass damit allgemein eine Abschätzung der Anzahl Bit-m-Operationen zur Zerlegung eines Gates in einem n -Bit-System möglich ist.

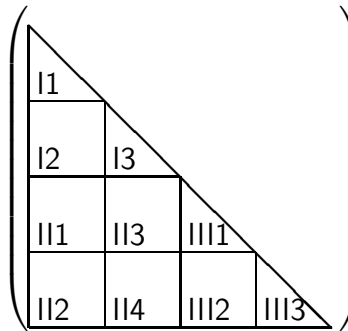
Satz 2.16 Sei ein Gate $M \in \mathcal{H}^{\otimes n, \otimes n}$ gegeben. Dann werden höchstens

$$n \cdot 2^n - 2^{n+1} + n + 2$$

Bit-m-Operationen zur Zerlegung von M benötigt.

Beweis. Die Grundidee ist wiederum gleich der allgemeinen Zerlegungs-idee, nämlich unterhalb der Diagonalen der $2^n \times 2^n$ -dimensionalen Matrix M Nullen zu erzeugen. Allerdings wird der

betreffende Bereich der Matrix zunächst folgendermaßen gegliedert.



Wird das obere Dreieck „(I)“ bestehend aus den Bereichen $I1$, $I2$ und $I3$ betrachtet, soll angenommen werden, dass es für diesen Bereich der Dimension $2^{n-1} \times 2^{n-1}$, welcher im Gesamten einem um ein Bit verkleinerten System entspricht, bereits eine Zerlegung gibt. Nützt man dort verwendete Bit- m -Operationen aus, um in dem quadratischen Bereich (II) - $II1$ bis $II4$ - Nullen zu erzeugen, kommt es zu Einsparungen bei der Zerlegung der Matrix M gegenüber der im Beweis zu Satz 2.15 verwendeten Zerlegung. Der Bereich unten rechts (III) - $III1$ bis $III3$ - entspricht in der Zerlegung dem oberen Bereich (I). Nach diesem Schema wurde bereits der verkleinerte Bereich zerlegt und es soll deshalb ein induktives Vorgehen angewendet werden. Zu beachten ist noch, dass bei der Zerlegung ein spaltenweises Vorgehen notwendig ist, um ein mögliches Zerstören bereits erzeugter Nullen zu verhindern, und es soll keine zwei aufeinanderfolgende Bit- m -Operationen auf dem selben Bit geben. Bei $n = 1$ wird für die Zerlegung einer 2×2 -Matrix lediglich eine Bit-1-Operation benötigt. Für $n = 2$ ist ein wenig mehr Aufwand zur Zerlegung der 4×4 -Matrix nötig. Zunächst werden mit einer Bit-2-Operation zwei Nullen in der linken Spalte erzeugt. Danach folgt eine Bit-1-Operation, die die dritte Null in der ersten Spalte und eine Null in der zweiten Spalte erzeugt. Anschließend kann die zweite Null in Spalte zwei mit einer Bit-2-Operation erstellt und zuletzt die letzte benötigte Null in der dritten Spalte (Bit-1-Operation) erzeugt werden. Somit werden vier Bit- m -Operationen gebraucht.

Induktion über n .

$$n = 1. \quad n \cdot 2^n - 2^{n+1} + n + 2 = 1 \quad \checkmark$$

$$n = 2. \quad n \cdot 2^n - 2^{n+1} + n + 2 = 4 \quad \checkmark$$

$n - 1 \rightarrow n$. Die Lösung sei für $n - 1$ richtig, d.h. für $(n - 1)$ q-Bits werden höchstens $x_{n-1} = (n - 1)2^{n-1} - 2^n + n + 1$ Bit- m -Operationen zur Zerlegung benötigt. Das Dreieck (I) entspricht dem Problem für $n - 1$ q-Bits und lässt sich mit x_{n-1} Operationen unterhalb der Diagonalen mit Nullen füllen. Sind 2^{n-1} Spalten mit Nullen besetzt, so bleibt das Dreieck (III) unten rechts mit ebenfalls x_{n-1} Operationen übrig. Damit gilt: $x_n = x_{n-1} + k + x_{n-1} = 2x_{n-1} + k$. Es bleibt zu zeigen: Die Anzahl zusätzlich benötigter Operationen beträgt $k = 2^n - n$ für Quadrat (II).

Für die Zerlegung des quadratischen Bereichs wird in jeder Spalte auf jeden Fall eine Bit- n -Operation benötigt. Die erste Spalte des quadratischen Bereichs (II) kann durch eine Bit- n -Operation mit Nullen besetzt werden. Da für jede Spalte im verkleinerten System (I) eine Bit- $(n-1)$ -Operation im dabei quadratischen Bereich $I2$ gebraucht wurde, können diese Operationen verwendet werden, um jeweils eine Spalte nach rechts versetzt vollständig Nullen im Bereich $III1$ zu erzeugen, bzw. mit der letzten solchen Operation die erste Spalte des Bereichs $III3$ mit Nullen

zu besetzen. Der Rest des Bereichs $II3$ wird dann zum jeweiligen Zeitpunkt mit einer zusätzlichen Bit-($n-1$)-Operation mit Nullen besetzt, also mit weiteren $2^{n-2} - 1$ Operationen. Nach jeder Spaltenoperation im Bereich $II1$ werden Operationen, die für den Bereich $I2$ gebraucht werden, analog für den Bereich $II2$ genutzt mit folgender Einschränkung: Es muss dabei darauf geachtet werden dass die Diagonale in $II2$ unberührt bleibt. Lediglich vor der nächsten Bit-($n-1$)-Operation für den Bereich $II1$ ist eine zusätzliche Bit- n -Operation für den Bereich $II2$ durchzuführen. Für die beiden Bereiche $II1$ und $II2$ werden insgesamt zusätzlich 2^{n-2} Operationen benötigt. Übrig bleibt der Bereich $II4$, der aber mittels $I3$ mit Nullen besetzt werden kann. Die Zerlegung erfolgt auf die selbe Weise, wie sie im System (I) für die Zerlegung von $I2$ mit Hilfe von $I1$ vorgenommen wurde. Lediglich statt der Bit-($n-1$)-Operationen müssen nun Bit- n -Operationen verwendet werden. Es werden so zusätzlich die für das reduzierte System zusätzlich benötigten Operationen auch hier angewendet. Das sind dann $x_{n-1} - 2x_{n-2}$ Operationen. Insgesamt ergeben sich

$$k = 2^{n-2} + 2^{n-2} - 1 + x_{n-1} - 2x_{n-2}$$

weitere Operationen. Für M werden damit insgesamt

$$x_n = 3x_{n-1} - 2x_{n-2} + 2^{n-1} - 1$$

Operationen benötigt. Einsetzen von x_{n-2} bzw. x_{n-1} (aus Induktionsannahme bekannt) in die Gleichung und Ausrechnen ergeben dann

$$x_n = n \cdot 2^n - 2^{n+1} + n + 2$$

Bit- m -Operationen und damit $k = 2^n - n$ zusätzliche Operationen zu x_{n-1} .

□

2.4 Zusammenfassung

Das Analogon zu klassischen Bits und den klassischen Änderungsmöglichkeiten der Bits bilden im QC die Multi-q-Bits und die Gates.

Einem klassischen Bit in einem n -Bit-System entspricht dabei ein reiner Zustand in Form eines orthonormierten Basisvektors im n -dimensionalen Hilbertraum $\mathcal{H}^{\otimes n}$ über \mathbb{C} . Ein Gate in Form einer unitären Matrix in $\mathcal{H}^{\otimes n, \otimes n}$ steht für die zeitliche Entwicklung des (konservativen) Systems.

Ein-Bit-Operationen, die die Wahrscheinlichkeitsverteilung ausschließlich für ein Bit ändern, heißen Bit- m -Operationen. In diese Klasse gehören ebenfalls alle Controlled-Operationen. Die bitunabhängige Strukturgleichheit der Bit- m -Operationen macht eine quantentheoretische Operation mit einer klassischen Berechnung vergleichbar. Durch die multiplikative Kombination einzelner Bit- m -Operationen können beliebige Gates konstruiert werden. Ebenso kann umgekehrt ein beliebiges Gate in eine multiplikative Abfolge von Bit- m -Operationen zerlegt werden. Die Anzahl der während der Zerlegung eines Gates maximal benötigten Bit- m -Operationen kann zwar gegenüber [Born03] reduziert werden, liegt allgemein aber immer noch im exponentiellen Bereich.

Aus quantentheoretischer Sicht benötigt die Anwendung eines Gates unabhängig von den Einträgen stets den gleichen Rechenaufwand einer Quantenoperation. Der Aufwand einer Matrix-mal-Vektor Berechnung hängt jedoch stark davon ab, wie viele Einträge eine Matrix besitzt. Durch die Bildung von Bit-m-Operationen besitzt jedes Gate immer höchstens zwei Spalten-einträge. Das bringt zwei große Vorteile: Erstens ist jede Berechnung eines Matrix-mal-Vektor Produktes mit Bit-m-Operationen mit dem gleichen Aufwand verbunden und zweitens ist damit ein Vergleichspunkt zwischen dem Aufwand für die Berechnung eines Quantenalgorithmus auf einem Quantenrechner und dessen Simulation auf einem klassischen Rechner gegeben. Auf dieser Grundlage kann nun im folgenden Abschnitt der Begriff der QA eingeführt werden.

KAPITEL 3

Quantenalgorithmen

Die heutige Rolle der QA besteht darin, gegenüber klassischen Verfahren einen Aufwandsgewinn vorzuweisen. Ein Aufwandsgewinn ist aber nur feststellbar, wenn sich Operationen im klassischen Bereich mit denen eines Quantenrechners vergleichen lassen. Da es noch keine geeigneten Quantenrechner gibt, muss es eine andere Art des Vergleiches geben. Der letzte Abschnitt hat in Form der Bit-m-Operationen für diese Vergleichbarkeit gesorgt.

In [Ahar03] wird gezeigt, dass es äquivalente Darstellungen der QA gibt. Auf der einen Seite sind die der mathematischen Modellierung entsprechenden Quantenschaltkreise (Abschnitt 3.1.1) und auf der anderen Seite sind die eher der physikalischen Realität näher kommenden Adiabatischen QA (Abschnitt 3.1.2). Eines jedoch ist allen diesen QA gleich: Sie haben eine festgelegte Struktur, die mit einem auf stochastischen Merkmalen aufgebautem Verfahren vergleichbar ist (Abschnitt 3.1.3).

Die Bandbreite der heute vorhandenen QA ist auf wenige Techniken beschränkt. Ein Hilfsmittel ist eine **Black-Box**. Dabei liefert die quantentheoretische Black-Box auf Anfrage Informationen, die zur Lösung eines Problems beitragen helfen. Ziel ist, mit möglichst wenigen Anfragen auszukommen und die Wahrscheinlichkeit bestimmter Basiskomponenten in die Höhe zu bringen. Die Klasse der amplitudenverstärkenden Algorithmen (Abschnitt 3.3.2) nutzt dieses Hilfsmittel aus. Eine weitere Technik besteht in der Anwendung der **Fourier-Transformation**, deren Algorithmen sich unter dem Begriff Hidden-Subgroup-Probleme (Abschnitt 3.3.1) zusammenfassen lassen. Chronologisch neuere Algorithmen arbeiten direkt mit verschiedenen Hamilton-Operatoren. Beispielsweise werden zur Lösung von Graphenproblemen Markov-Ketten ähnliche Quantum-(Random)-Walks (Abschnitt 3.3.3) benutzt.

3.1 Charakteristika von Quantenalgorithmen

Wie in Abschnitt 1.4 über die Kopenhagener Deutung bereits erwähnt, ist ein QA aus drei Schritten (vgl. [Hebe00],[Heis79]) zusammengesetzt. Zunächst muss das System mit einem Startzustand vorbereitet werden. Davon ausgehend vollzieht sich die zeitliche Entwicklung des Systems, währenddessen sich der Startzustand nach und nach ändern kann. Zuletzt wird eine Messung des Systems vorgenommen. Das bedeutet, dass das System sich bezüglich einer festgelegten Zahl von q -Bits in einen determinierten Zustand bringen muss. Ein gemessenes q -Bit hat dann absolut den Wert 0 oder 1, es existiert keine Wahrscheinlichkeitsverteilung mehr für dieses q -Bit und dessen Superposition ist zerstört (vgl. Abschnitt 2.1.1).

Definition 3.1 *Ein Quantenalgorithmus ist zusammengesetzt aus drei Schritten:*

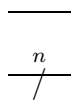
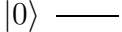
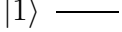
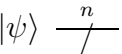
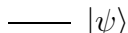
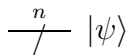


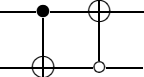

1. *Präparation des Systems in einen Startzustand*
2. *Zeitentwicklung des Systems*
3. *Messung des Systems*

Ein gemessenes q-Bit befindet sich dann in einem Eigenzustand nach (1.2) des Systems. Daher müssen Algorithmen so beschaffen sein, dass die Möglichkeiten der Superposition während der zeitlichen Entwicklung ausgenutzt werden, damit das zu messende Endergebnis mit höchster Wahrscheinlichkeit das gewünschte Resultat liefert. Jede Messung kann aber auch als echter Zufallszahlengenerator gebraucht werden (vgl. Beispiel 3.1.1). Die zeitliche Entwicklung ist das Merkmal, mit dem sich die verschiedenen Algorithmen unterscheiden lassen. Über die in Kapitel 1 beschriebenen Hamilton-Operatoren wird die Zeitentwicklung des Systems zeitkontinuierlich. In Kapitel 2 ist die zeitliche Entwicklung des Systems diskret als serielle Anwendung unitärer Matrizen beschrieben. Möglichkeiten zur Diskretisierung von QA werden in Abschnitt 3.2 gegeben.

3.1.1 Zeitdiskretes Schaltkreis-Modell

Physikalische Systeme werden üblicherweise durch ihren Hamilton-Operator beschrieben und entwickeln sich natürlicherweise in kontinuierlicher Zeit. Dargestellt wird diese physikalische Entwicklung über die Schrödinger-Gleichung gemäß (1.7). Ein zeitdiskreter QA wird beschrieben durch die seriell ausgeführte Anwendung von Gates auf einen gegebenen Startzustand. Mit anderen Worten ist ein QA zeitdiskret, sobald gemäß (1.15) in Postulat 3 der Zeitentwicklungsoperator unitär ist. Ein klassischer Computer arbeitet mit einer Menge von Eingabe-Bits und liefert als Ergebnis eine Menge von Ausgabe-Bits. Der ganze Eingabe-Verarbeitung-Ausgabe-Prozess kann als logischer Schaltkreis beschrieben werden. Das **Schaltkreis-Modell** korrespondiert eng mit der Philosophie des QC. Das Modell ist eine andere Darstellung für die zeitliche Entwicklung eines Systems, die einem logischen Schaltbild ähnlich ist. Allerdings gibt es hierbei eine festgelegte Struktur von links nach rechts (siehe etwa [Baue02],[Qcir@]). Es beginnt gemäß Definition 3.1 links mit dem Startzustand, gefolgt von der Abfolge der Gates und endet mit einem Ausgangszustand, der aufgrund einer Messung entsteht. Wie bei einem elektronischen Schaltbild steht eine begrenzte Anzahl von Schaltkreissymbolen zur Verfügung. Tafel 3.1 zeigt die für diese Arbeit benötigten Symbole. Eine umfangreichere Sammlung ist beispielsweise bei [Qcir@] zu finden.

Jeder zeitdiskrete QA nach Definition 3.1 lässt sich als Quantenschaltkreis mit Hilfe des Schaltkreis-Modells darstellen.

Zeichen	Bedeutung
	Leitung, die ein q-Bit bzw. ein Multi-q-Bit (mit übergestellter Anzahl der q-Bits) von links nach rechts „führt“.
$ 0\rangle$  $ 1\rangle$  $ \psi\rangle$ 	Reiner Startzustand eines q-Bits, entweder $ 0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ oder $ 1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ sind möglich. Ein Multi-q-Bit $ \psi\rangle$ wird durch das übergestellte n gekennzeichnet und steht für einen reinen Zustand eines n -Bit-Systems.
 	Endzustand eines q-Bits bzw. Multi-q-Bits.
	Gate U .
	Messung eines q-Bits.
	Besondere Gates: Controlled Bit-Flip, „CNOT“. Das kontrollierte q-Bit ist als ausgefüllter ($\equiv 1$) bzw. als nicht ausgefüllter ($\equiv 0$) Kreis zu sehen.
	Swap-Gate. Realisierbar durch höchstens $2n - 1$ Bit-m-Operationen.

Tafel 3.1: Ausgewählte Symbole eines Quantenschaltkreises

Definition 3.2 ((Zeitdiskreter) Quantenschaltkreis) Sei

$$U = U_m \cdot \dots \cdot U_1 \tag{3.1}$$

die diskrete zeitliche Entwicklung eines Systems mit unitären Matrizen U_k , $k = 1, \dots, m$. Dann stellt

$$|\psi_0\rangle \xrightarrow{n} \boxed{U} \xrightarrow{n} |\psi\rangle$$

den **Quantenschaltkreis** des zeitdiskreten Systems dar. Mit der zusätzlichen Durchführung einer Messung wird der Quantenschaltkreis mit dem Startzustand ψ_0 zu einem QA nach Definition 3.1.

Beispiel 3.1.1 Der folgende Quantenschaltkreis zeigt einen Zufallsgenerator für ein q -Bit.

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

Das Gate H ist dabei das so genannte **Hadamard-Gate**

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

Eine durchgeführte Messung am Ende ergäbe dann mit jeweils Wahrscheinlichkeit $\frac{1}{2}$ einen der beiden reinen Zustände. Wird die Anordnung erweitert auf n q -Bits, so entsteht der Zufallsgenerator

$$|\psi_0\rangle \xrightarrow{n} \boxed{H_n} \xrightarrow{n} \frac{\pm 1}{\sqrt{2^n}} \sum_{i=1}^{2^n} |\psi_i\rangle$$

durch jeweils einmalige Anwendung des Hadamard-Gates auf jedes einzelne q -Bit.

3.1.2 Adiabatische Quantenalgorithmen

QA, die auf Hamilton-Operatoren basieren, werden **Adiabatische Quantenalgorithmen** (vgl. [Chil00]) genannt. Dabei wird ein zeitabhängiger Hamilton-Operator, dessen Grundzustand auf einfache Weise (physikalisch) erzeugbar ist nach und nach in einen Hamilton-Operator verändert, wobei der Zustand des Systems dann dem gesuchten Zustand entspricht. Der Grundzustand eines Hamilton-Operators ist der Eigenvektor, dem der kleinste Eigenwert entspricht. In der Literatur (wie etwa auch bei [Chil00]) wird überwiegend eine lineare Interpolation vorgeschlagen. Der Start-Operator H_S wird zu dem Problem-Operator H_P gemäß [Kieu01@] verändert über

$$\tilde{H}(s) \stackrel{def}{=} (1 - s(t))H_B + s(t)H_P, \quad s = \frac{t}{T}, \quad t \in [0; T]$$

Gestartet wird bei $t = 0$. Für T geeignet groß und $t \rightarrow T$ wird der gesuchte Zustand mit hoher Wahrscheinlichkeit erreicht ([Kieu01@]). Ein Beispiel für einen Adiabatischen QA ist der Algorithmus von Grover, der folgendermaßen formuliert werden kann: Sei n die Anzahl der q -Bits, $|\psi\rangle$ die uniforme Superposition der Basiszustände und $|\psi_0\rangle$ der markierte Zustand. Dann kann der Grover-Algorithmus (vgl. (3.3.2)) über

$$H(s) = (1 - s)(I - |\psi\rangle\langle\psi|) + s(I - |\psi_0\rangle\langle\psi_0|)$$

formuliert werden (siehe [Lato03@]). Adiabatische QA sind zunächst einmal zeitkontinuierlich formuliert. Mit Hilfe der interpolierten Hamiltonoperatoren lassen sich die Algorithmen konstruieren - als ein Beispiel wurde der Grover-Algorithmus aufgeführt. Andererseits müssen die so konstruierten kontinuierlichen Algorithmen dann mit den Mitteln in Abschnitt 3.2 diskretisiert werden. Entscheidend ist die folgende in [Ahar03] bewiesene Aussage:

Satz 3.3 (Äquivalenz von Quantenschaltkreis und Adiabatischem QA)

Quantenzustände lassen sich genau dann effizient als Quantenschaltkreis realisieren, wenn sie sich als Adiabatische QA formulieren lassen.

3.1.3 Betrachtung als stochastischer Algorithmus

Ein zeitdiskreter QA ist als serielle Anwendung unitärer Matrizen auf einen Startbasisvektor mit einer abschließenden Messung beschreibbar. Diese Abfolge ist beliebig oft wiederholbar und bei jeder Wiederholung ist durch die Messung ein anderes Ergebnis möglich. Erst durch eine häufige Wiederholung wird ein bestimmtes Muster sichtbar, indem die Ergebnisse in der Häufigkeit ihrer Wahrscheinlichkeit eintreten. Diese Charakteristik führt unmittelbar zu dem Begriff des stochastischen Verfahrens. Es wird sich herausstellen, dass es sich - wie in [Math04] bereits erwähnt - bei einem QA um keinen stochastischen Algorithmus handelt. Für die formale Beschreibung des Experimentes werden die grundlegenden Definitionen 1 bis 5 benötigt. Der Ausgangspunkt sei der Wahrscheinlichkeitsraum

$$(\Omega, \mathcal{P}(\Omega), P),$$

wobei $\Omega = \{e_1, \dots, e_{2^n}\}$ und P das durch den Zustandsvektor ψ des Systems induzierte Wahrscheinlichkeitsmaß ist. Sei nun in einem n -Bit-System ein QA gegeben, $U = U_m \cdot \dots \cdot U_1 \in \mathcal{H}^{2^n, 2^n}$ die diskrete zeitliche Entwicklung des Systems mit unitären Matrizen U_k , $k = 1, \dots, m$ gemäß (3.1) und sei $e_i \in \mathbb{R}^{2^n}$ der gewählte Startvektor. Betrachtet man

$$U \cdot e_i = (U_m \cdot \dots \cdot U_1) \cdot e_i = u_i = \begin{pmatrix} u_{1i} \\ \vdots \\ u_{2^n i} \end{pmatrix} \quad (3.2)$$

mit Einträgen u_{ij} in der i -ten Zeile und j -ten Spalte von U , so wird mit einem Wahrscheinlichkeitsraum $(\tilde{\Omega}, \tilde{\mathcal{A}}, \tilde{P})$

$$X : \tilde{\Omega} \rightarrow \Omega \quad (3.3)$$

zu einer Zufallsvariablen und es gilt

$$P(\{e_j\}) = \tilde{P}_X(\{e_j\}) = \tilde{P}(\{\omega \in \tilde{\Omega} : X(\omega) = e_j\}) = |u_{ji}|^2.$$

Hierbei wird also ein Gate U auf den Startvektor e_i angewendet. Die Frage ist nun, ob die Betrachtung der Anwendung der Serie einzelner Matrizen U_k auf einen Vektor der Länge 1 als algorithmische Abfolge von Zufallsvariablen gesehen werden kann.

Definition 3.4 (stochastischer Prozess) Seien $(\tilde{\Omega}, \tilde{\mathcal{A}}, \tilde{P})$ ein Wahrscheinlichkeitsraum und (Z, \mathcal{A}_Z) ein Messraum. Eine Folge von Zufallsvariablen $(X_t)_{t \in I}$, I diskrete endliche oder unendliche Indexmenge, mit $X_t : \tilde{\Omega} \rightarrow Z$ heißt **zeitdiskreter stochastischer Prozess mit Zustandsraum Z** .

Der Begriff der Abfolge von Zufallsvariablen in Algorithmen wie etwa beim Metropolis-Algorithmus (vgl. [Suto03]) wird mit dem Begriff des **stochastischen Algorithmus** beschrieben. Die Zufallsvariablen sollen der Markov-Eigenschaft

$$P(X_{t+1} = j | X_t = i_t) = P(X_{t+1} = j | X_0 = i_0, X_1 = i_1, \dots, X_t = i_t) \quad (3.4)$$

genügen. Ausgehend von der Problemstellung (3.2) sollen die unitären Matrizen der Reihe nach auf den jeweiligen Zustandsvektor angewendet werden (vgl. [Math04]). Sei $u_{ij}^{(k)}$ der Eintrag in der i -ten Zeile und j -ten Spalte der Matrix U_k . Im ersten Schritt wird die Matrix U_1 auf den festgelegten Startvektor e_i multipliziert. Das kann durch eine Zufallsvariable

$$X_1 : \tilde{\Omega} \rightarrow \Omega \quad (3.5)$$

modelliert werden. Die Wahrscheinlichkeit, dass die Zufallsvariable den Wert e_j annimmt, ist

$$P(\{e_j\}) = \tilde{P}_{X_1}(\{e_j\}) = |u_{ji}^{(1)}|^2.$$

Allgemein ergibt sich im $t + 1$ -ten Schritt des Algorithmus die Zufallsvariable

$$X_{t+1} : \tilde{\Omega} \rightarrow \Omega. \quad (3.6)$$

Unter der Annahme der Realisierung eines reinen Zustandes $X_t(\omega \in \tilde{\Omega}) = e_i$ im t -ten Schritt lässt sich die Übergangswahrscheinlichkeit

$$p_{ij}^{(t+1)} = |u_{ji}^{(t+1)}|^2 \quad (3.7)$$

von e_i nach e_j im Schritt $t + 1$ angeben. Die Realisierung setzt aber in der Betrachtung des QA eine Messung voraus. Diese ist aber nicht gegeben. Dennoch lassen sich für den $t + 1$ -ten Schritt Wahrscheinlichkeiten für eine gedachte Realisierung eines reinen Zustandes bestimmen. Da nicht von einer Realisierung im t -ten Schritt ausgegangen werden darf, muss dann das diskrete Wahrscheinlichkeitsmaß (d.h. die Verteilung) vorliegen. Im QA wird die Matrix U_{t+1} angewendet auf den durch die Berechnung von

$$\lambda^{(t)} \stackrel{def}{=} U_t \cdot \dots \cdot U_1 \cdot e_i$$

entstandenen Vektor $\lambda^{(t)}$. Es ergibt sich $\tilde{P}_{X_t}(\{e_j\}) = |\lambda_j^{(t)}|^2$, $j = 1, \dots, 2^n$ und damit lässt sich im $t + 1$ -ten Schritt $\lambda_i^{(t+1)}$ berechnen als

$$\lambda_i^{(t+1)} = \sum_{j=1}^{2^n} \lambda_j^{(t)} u_{ij}^{(t+1)}, \quad i = 1, \dots, 2^n.$$

Das bestimmt die Wahrscheinlichkeit

$$\tilde{P}_{X_{t+1}}(\{e_j\}) = |\lambda_j^{(t+1)}|^2, \quad (3.8)$$

dass die Zufallsvariable X_{t+1} den Wert e_j annimmt und legt die neue Verteilung der Zufallsvariablen X_{t+1} fest.

Bemerkung. Die Verteilung von X_{t+1} hängt von der Verteilung von X_t ab. Die Realisierung von X_t bedeutete einen immensen Informationsverlust und lieferte eine andere Verteilung von X_{t+1} . Deswegen handelt es sich bei einem QA, der aus mehr als einem Gate besteht, nicht um einen stochastischen Algorithmus im klassischen Sinne mit der Eigenschaft (3.4). Überlegt man

außerdem, dass die Verteilung von X_t von der Verteilung von X_{t-1} abhängt usw., so wird klar, dass in einem QA bei der Berechnung auf klassischen Rechnern eine ungeheure Komplexität liegt.

Wird jedoch nach jedem Schritt eine Realisierung d.h. Messung eines reinen Zustandes durchgeführt, ist die Wahrscheinlichkeit, dass im Schritt $t + 1$ der reine Zustand e_j realisiert wird,

$$\tilde{P}_{X_{t+1}}(\{e_j\}) = \tilde{P}_{X_{t+1}}(X_{t+1} = e_j | X_t = e_{i_t}, X_{t-1} = e_{i_{t-1}}, \dots, X_1 = e_{i_1}).$$

Nach Messung des reinen Zustandes e_{i_t} in X_t findet eine Zustandsreduktion nach Abschnitt 2.1.1 statt und es entsteht gemäß (2.3) das neue Multi-q-Bit e_{i_t} . So kann eine Übergangsmatrix mit den Übergangswahrscheinlichkeiten in (3.7) angegeben werden und die Verteilung für X_{t+1} hängt letztlich nur von der Realisierung der vorhergehenden Zufallsvariablen ab,

$$\tilde{P}_{X_{t+1}}(\{e_j\}) = \tilde{P}_{X_{t+1}}(X_{t+1} = e_j | X_t = e_{i_t}).$$

Damit ist ein stochastischer Algorithmus mit den Zufallsvariablen X_j , $j = 1, \dots, k$ entstanden. Denn die Verteilung von X_j hängt nun lediglich von einer Realisierung von X_{j-1} ab. Der letzte Gedanke ist die Grundlage des in Abschnitt 4 dargestellten Verfahrens zur Approximation des Ergebnisses eines QA. Dabei wird dann im stochastischen Algorithmus nach der Realisierung von e_j im Schritt $t + 1$ ein zur Folge aller Realisierungen gehörender Linearfaktor mitgeführt, der rekursiv bestimmbar ist (vgl. Kapitel 4):

$$\lambda_j^{(t+1)} = \lambda_{i_t}^{(t)} u_{j i_t}^{(t)}. \quad (3.9)$$

Dabei gibt es nach jedem Schritt $t+1$ einen Informationsverlust $I_{verl,t+1}$. Die gesamte Information setzt sich zusammen aus der noch vorhandenen Information $I_{vorh,t+1}$ und der verloren gegangenen Information $I_{verl,t+1}$. $I_{verl,t+1}$ und $I_{vorh,t+1}$ können nach jedem Schritt $t + 1$ des Algorithmus quantifiziert werden durch

$$\begin{aligned} I_{verl,t+1} &= I_{verl,t} + I_{vorh,t} \cdot (1 - |\lambda_j^{(t+1)}|^2) \\ I_{vorh,t+1} &= I_{vorh,t} \cdot |\lambda_j^{(t+1)}|^2 = 1 - I_{verl,t+1}. \end{aligned} \quad (3.10)$$

Nach einem algorithmischen Durchlauf ist nur noch ein Bruchteil der Gesamtinformation vorhanden. Durch sukzessive Zugabe von Information in Form zusätzlicher Realisierungen kann die volle Information und damit auch volle Komplexität erreicht werden. In Abbildung 1 wurde bereits ein schematischer Verlauf für den Vergleich von Komplexität und Informationsverlust gezeigt.

3.2 Diskretisierung von Quantenalgorithmen

Bestimmte QA liegen als zeitkontinuierliche Modelle in Form einer Beschreibung über einen Hamilton-Operator vor (z.B. Adiabatische QA). Damit diese aber mit Hilfe eines Schaltkreis-Modells und damit mit dem Branch&Bound-Verfahren in Kapitel 4 gelöst werden können, müssen sie in ein zeitdiskretes Modell „übersetzt“ werden. Sie müssen approximiert werden. Doch nur

bestimmte Hamilton-Operatoren können effektiv durch unitäre Matrizen approximiert werden. In [Chil00] finden sich einige Techniken, mit welchen diese Approximation durchgeführt werden kann, und folgende Definition, wann eine Approximation möglich ist.

Definition 3.5 (Effektive Approximation von Hamilton-Operatoren) Gegeben sei ein Hamilton-Operator H . Gilt dann für ein Produkt $U = U_k \cdot \dots \cdot U_1$ unitärer Matrizen $\|U - e^{-iHt}\| < \epsilon$ für ein $t > 0$ und $\epsilon > 0$, so wird H durch U **effektiv approximiert**.

Das gilt sowohl für zeitabhängige als auch für zeitunabhängige Hamilton-Operatoren. Bei der Simulation von zeitabhängigen Hamilton-Operatoren ist zunächst zu beachten, dass sie diskretisiert werden müssen, bevor sie approximiert werden können. Da diese Problematik bei Adiabatischen QA, die in dieser Arbeit nicht detailliert betrachtet werden, auftritt, sollen nur ohne Beweis zwei Techniken zur effektiven Approximation zeitunabhängiger Hamilton-Operatoren betrachtet werden, die für die später im Kapitel 5 folgenden Simulationen benötigt werden.

Sei

$$H = \sum_{k=1}^n H_k \quad (3.11)$$

der Hamilton-Operator eines Systems nach (1.7). Dann gibt es folgende in [Chil00] bewiesene Techniken (für ein $j \in \{1, \dots, n\}$) zur Approximation von Hamilton-Operatoren durch unitäre Matrizen.

- **Unitäre Transformation.** Kann H effektiv approximiert werden, so auch UHU^\dagger , falls U unitär ist. Es ergibt sich

$$e^{-iUHU^\dagger t} = U e^{-iHt} U^\dagger. \quad (3.12)$$

- **Addition.** Sind alle H_k , $k \in \{1, \dots, n\}$, effektiv approximierbar, so auch H . Es ergibt sich

$$e^{-iHt} = e^{-i \sum_{k=1}^n H_k t} = \lim_{m \rightarrow \infty} (e^{-iH_1 t/m} \cdot \dots \cdot e^{-iH_n t/m})^m \quad (3.13)$$

Um einen möglichst kleinen Fehler ($\epsilon > 0$) zu erhalten, sollte m bei einer Approximation in der Größenordnung

$$\mathcal{O}\left(\frac{n}{2\epsilon} (t \cdot \max_{j_1, j_2} \|[H_{j_1}, H_{j_2}]\|)^2\right)$$

gewählt werden. Kommutieren die H_j gemäß (1.6), so gilt

$$e^{-iHt} = e^{-i \sum_{k=1}^n H_k t} = e^{-iH_1 t} \cdot \dots \cdot e^{-iH_n t}$$

Bemerkung. Feynmann zeigte in [Feyn85] die umgekehrte Richtung: Zu einem gegebenen Schaltkreis-Modell gibt es einen zeitunabhängigen Hamilton-Operator der Form (3.11). Damit ist jeder Hamilton-Operator der Form (3.11) eine dem Schaltkreis-Modell äquivalente Darstellung, wie es bereits in Satz 3.3 besprochen wurde.

3.3 Klassifikation von Quantenalgorithmen

Zu Beginn des Kapitels wurden drei verschiedene Klassen erwähnt und sollen im Hinblick auf das später erläuterte Branch&Bound-Verfahren etwas genauer¹ untersucht werden. Zwei Merkmale bestimmen die drei Klassen: Einerseits ist jeder Klasse eine nur in ihr vorkommende Prozedur im QA zu eigen, andererseits gehen die Algorithmen jeder Klasse auf eigene Problemstellungen zurück. Dies soll in den folgenden Abschnitten herausgearbeitet werden. Insbesondere die zur Anwendung kommende Methodik spielt bei der Analyse der Güte des Verfahrens in Kapitel 4 eine wichtige Rolle.

3.3.1 Hidden Subgroup-Probleme

Zu den ersten entwickelten QA zählen unter einen Begriff subsummiert die „Hidden Subgroup-Probleme“. In polynomialer Zeit wird die Periodizität von Funktionen gefunden. Die Problemstellung kann gemäß [Lomo02@] formuliert werden.

Definition 3.6 (Hidden Subgroup-Struktur) Eine Abbildung $\phi : A \rightarrow S$ einer Gruppe A in eine Menge S besitzt eine **Hidden Subgroup-Struktur**, wenn eine Untergruppe (**Hidden Subgroup**) K_ϕ von A und eine Abbildung $i_\phi : A/K_\phi \rightarrow S$ existieren, sodass das Diagramm

$$\begin{array}{ccc} A & \xrightarrow{\phi} & S \\ \nu \searrow & & \nearrow i_\phi \\ & A/K_\phi & \end{array}$$

kommutativ ist, wobei A/K_ϕ die Sammlung rechter Co-Mengen von K_ϕ in A und $\nu : A \rightarrow A/K_\phi$ die natürliche Abbildung von A nach A/K_ϕ sind. Ist K_ϕ eine normale Untergruppe von A , dann ist A/K_ϕ eine Quotientengruppe und ν ist ein Epimorphismus.

Damit lässt sich das Hidden Subgroup-Problem definieren.

Definition 3.7 (Quantum Hidden Subgroup-Algorithmus) Sei $\phi : A \rightarrow S$ eine Abbildung mit Hidden Subgroup-Struktur. Ein Algorithmus, der eine Hidden Subgroup K_ϕ findet, wird **Hidden Subgroup-Algorithmus** genannt. Ein QA, der bei gegebener unitärer Transformation $U_\phi : \mathcal{H}_A \times \mathcal{H}_S \rightarrow \mathcal{H}_A \times \mathcal{H}_S$ eine Hidden Subgroup K_ϕ mit beschränkter Fehlerwahrscheinlichkeit und mit wenigen Black-Box Anfragen an U_ϕ findet, heißt **Quantum Hidden Subgroup-Algorithmus**.

¹Für eine über die erwähnten Zusammenhänge hinaus gehende Untersuchung wird auf die referenzierte Literatur verwiesen.

Ohne näher auf die speziellen Algorithmen einzugehen, lässt sich feststellen (siehe [Mosc99@]): Alle Hidden Subgroup-Algorithmen haben wichtige Bestandteile gemeinsam: Zunächst die Präparation einer uniformen Superposition zu Beginn mit Hilfe von Hadamard-Gates und die Prozedur der „Quantum Fourier Transformation“ (QFT). Ein Problem wird in ein anderes Problem transformiert und dafür ist vergleichsweise einfach eine Lösung zu finden. Durch das Ausnutzen der Superposition lassen sich im Bereich des QC bestimmte Transformationen schneller durchführen als auf klassischem Wege. Dazu zählt auch die diskrete Fourier Transformation (dFT) ([Niel00]). Um die Superposition des Multi-q-Bits nach Ausführung der QFT nicht zu zerstören kann die QFT stets nur als Baustein eines QA gesehen werden. Mit Hilfe dieses Bausteines ist es möglich (vgl. [Lomo02@]), einen erfolgreichen Quantum Hidden Subgroup-Algorithmus zu bauen. Die QFT entspricht der klassischen dFT und wird folgendermaßen auf ein Multi-q-Bit angewendet:

Definition 3.8 (Quantum Fourier Transformation) Seien mit e_j , $j = 1, \dots, 2^n$ die Basiszustände aus n q-Bits ausgezeichnet. Dann heißt die lineare Transformation QFT aller Basiszustände mit

$$\text{QFT} : e_j \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=1}^{2^n} e^{2\pi i(j-1)(k-1)/2^n} e_k \quad (3.14)$$

Quantum Fourier Transformation.

Ein beliebiges Multi-q-Bit gemäß (2.1) wird dann entsprechend

$$\text{QFT}\left(\sum_j \lambda_j e_j\right) = \sum_j \lambda_j \text{QFT}(e_j)$$

transformiert.

Bemerkung. Eine wichtige Eigenschaft der QFT ist die Unitarität ([Lomo02@]). Ohne diese Eigenschaft wäre es nicht möglich, die QFT in QA einzusetzen.

Die QFT ist als Produkt einzelner unitärer Transformationen darstellbar und kann so als Schaltkreis-Modell dargestellt werden (vgl. [Niel00]). Dazu werden noch folgende Matrizen benötigt, die dann, wie unten im Schaltkreisbild 3.1 der QFT zu sehen, als Controlled-Operationen in den Schaltkreis integriert werden müssen:

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

Beispiele für Quantum Hidden Subgroup-Algorithmen sind die QA von Deutsch, Deutsch-Josza, Simon und Shor [Lomo02@], die jeweils auf abelschen Gruppen arbeiten ([Lomo04@]), Abelian Stabiliser Probleme [Kita95@] oder auch Hallgrens QA zur Lösung von Pells Gleichung [Josz03@], u.a.

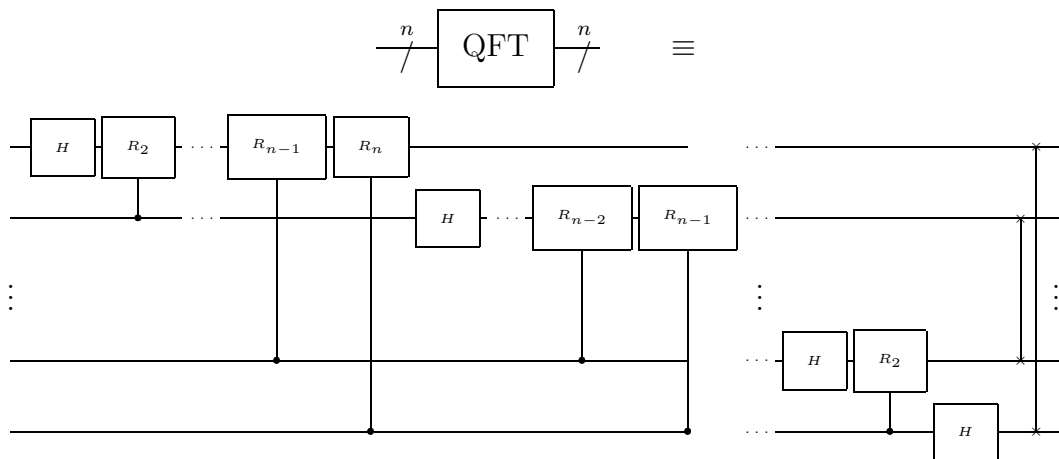
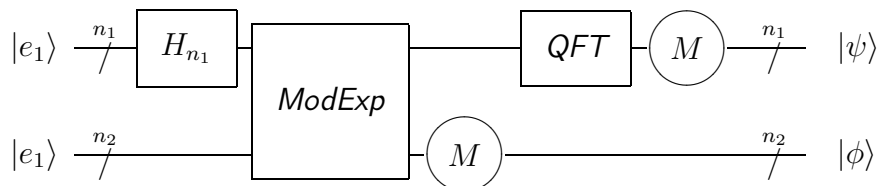


Bild 3.1: QFT in Form von Bit-m-Operationen.

Beispiel 3.3.1 (Shors Faktorisierungsalgorithmus) Zu einer vorgegebenen natürlichen Zahl $z \in \mathbb{N}$ kann mit einem QA eine Produktdarstellung $z = p \cdot q$ mit Primfaktoren p und q gefunden werden. Die Anzahl der dafür benötigten q -Bits richtet sich nach der zu faktorierenden Zahl z . Es werden zwei Register angelegt. Das Erste enthält n_1 , $z^2 \leq 2^{n_1} \leq 2 \cdot z^2$, das Zweite $n_2 = \lceil \log_2(z) \rceil$ q -Bits und insgesamt somit ein Multi- q -Bit aus $n_1 + n_2$ - q -Bits. Das Erste wird in uniforme Superposition gebracht, danach wird auf beiden Registern die modulare Exponentiation durchgeführt. Nach Messung des zweiten Registers wird auf dem Ersten die QFT angewendet und zum Schluss wird das Erste gemessen (vgl. [Aume03@]):



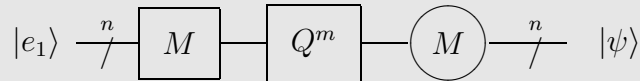
3.3.2 Amplitudenverstärkung

Sei \mathcal{H} der Hilbertraum eines Quantensystems. Eine Boolesche Funktion $f : \mathbb{Z} \rightarrow \{0, 1\}$ teilt den Ergebnisraum in zwei Unterräume. Dabei entspricht die „1“ dem gesuchten Unterraum. Jeder beliebige Zustand aus \mathcal{H} besitzt eine Wahrscheinlichkeit dafür, dass der gesuchte Zustand gemessen wird. Der folgende unitäre Operator erhöht die Amplitude für die gesuchten Zustände, d.h. die Wahrscheinlichkeit einer Messung der gesuchten Zustände (siehe [Bras00@]).

$$Q \stackrel{def}{=} -MU_0M^\dagger U_f$$

U_0 ändert das Vorzeichen der Amplitude des ersten Basiszustandes, U_f ändert das Vorzeichen von $x \in \mathbb{Z}$ genau dann, wenn $f(x) = 1$ gilt. M muss unitär sein.

Satz 3.9 Sei M ein Quantenschaltkreis. Sei $f : \mathbb{Z} \rightarrow \{0, 1\}$ eine Boolesche Funktion. Sei a die Anfangswahrscheinlichkeit für das Auffinden des gesuchten Ergebnisses. Setze $m = \lfloor \frac{\pi}{4\theta_a} \rfloor$ mit $a = \sin^2(\theta_a)$, $0 \leq \theta_a \leq \frac{\pi}{2}$. Dann ergibt $Q^m M e_1$



nach einer Messung mit Wahrscheinlichkeit

$$\max(1 - a, a)$$

den gesuchten Zustand.

Der Beweis findet sich bei [Bras00@].

Beispiel 3.3.2 (Suchalgorithmus von Grover) Bei der Suche von Datensätzen in einer unstrukturierten Datenbank - wie etwa der Suche einer Telefonnummer in einem Telefonbuch - wurden Quanten-Suchalgorithmen gefunden, die einen quadratischen Speedup gegenüber klassischen Suchverfahren bieten. Mathematisch heißt das, es gibt eine Funktion $f : \{0, 1\}^n \rightarrow \{0, 1\}$, wobei es wenige Werte a_i , $i = 1, \dots, k$ mit $f(a_i) = 1$ gibt. Für alle übrigen Werte gelte $f(b) = 0$, $b \neq a_i \quad \forall i$. Eine geeignete Quanten-Black-Box U_f liefert nun genau den Funktionswert von f für einen Eingabewert zurück. Die Aufgabe ist nun, mit möglichst wenigen Anfragen einen der gesuchten Werte a_i zu finden. Dies ist nach Grover (vgl. [Niel00]) mit $M = H_n$ (Hadamard auf n Bits) in $m \sim \mathcal{O}(\sqrt{\frac{2^n}{k}})$ Anfragen mit dem Quantenschaltkreis gemäß Satz 3.9 zu lösen.

Bemerkung. Entscheidend für das Bestimmen des Ergebnisses bei den amplitudenverstärkenden Algorithmen ist demnach die unitäre Matrix M . Ohne a-priori Informationen über die gesuchten Basisvektoren wird ein Zufallsgenerator für die Matrix M gewählt. Beim Algorithmus von Grover ist dies genau der Fall, da jeder mögliche Zustand zunächst mit gleicher Wahrscheinlichkeit angenommen werden kann.

3.3.3 Quantum-(Random)-Walk

Randomisierte Algorithmen und Monte-Carlo Markov-Ketten sind in der Praxis sehr erfolgreich ([Kemp04]). Solche Markov-Ketten sind so in QA umzusetzen, dass die stationäre Verteilung das Ergebnis des Problems ist. Ein **Random-Walk** ist zu verstehen als Wanderung entlang der Knoten eines durch Kanten verbundenen Graphen. Die eingeschlagene Richtung wird durch probabilistische Vorgaben entschieden. Der Übergang von einem Knoten zum Anderen kann sowohl zu ganz bestimmten diskreten als auch zu beliebigen kontinuierlichen Zeitpunkten geschehen. Dieser Vorgang ist eine Markov-Kette bzw -Prozess.

Quantum-(Random)-Walks garantieren keinen algorithmischen Speedup - siehe [Chil02@]. Erst der in [Chil02@] eingeführte **Quantum Walk**, der ein modifizierter Quantum-(Random)-Walk ist,

löst das dabei gegebene Problem subexponentiell im Gegensatz zu klassischen Algorithmen. Ausgehend von der zeitkontinuierlichen Definition kann der Algorithmus mit den Mitteln in Abschnitt 3.2 diskretisiert werden. Eine Besonderheit des hier vorgestellten Quantum Walks ist noch, dass für die kontinuierliche und damit auch für die diskrete Version der zu den Knoten des gegebenen Graphen korrespondierende Hilbertraum um extra „q-Bits“ erweitert werden muss.

Quantum-Walk auf zwei Binärbäumen

Benötigt wird ein **Graph** G aus N Knoten und einer Kantenmenge, die beschreibt, welche Knoten miteinander verbunden sind. Zwei der Knoten sind speziell markiert: Der Startknoten („START“) und der gesuchte Ausgangsknoten („ENDE“). Der für den Quantum-Walk spezielle Graph G_k wird aus zwei Graphen G in Form von Binärbäumen der Tiefe k und jeweils 2^k Blättern zusammengesetzt. Die Blätter der beiden Bäume seien ebenfalls über Kanten verbunden. Ein willkürlich gewähltes Blatt des einen Baumes werde per Zufall mit einem Blatt des anderen Baumes verbunden. Dieses Blatt werde seinerseits wiederum zufällig mit einem Blatt des ersten Baumes verbunden, usw. bis jedes Blatt mit zwei anderen Blättern des zweiten Baumes verbunden ist. Bild 3.2 zeigt einen solchen Graphen für $k = 4$. Der am weitesten links liegende Knoten ist

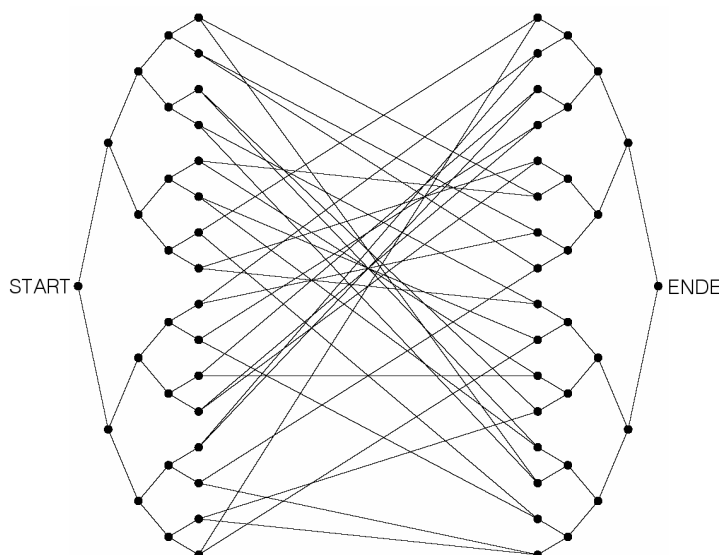


Bild 3.2: Beispiel eines Graphen G_4

der START- und der am weitesten rechts liegende der ENDE-Knoten. In [Chil02@] wird gezeigt, dass genau diese Problemstellung als Quantum-Walk beschrieben und in polynomialer Zeit gelöst werden kann im Gegensatz zu einem klassischen Algorithmus. Das führt zu folgender

Problemstellung. Sei eine Black-Box für einen Graphen und dessen START bekannt. Finde ENDE.

Sei $a \in G$ ein Knoten eines Graphen G und $a\tilde{a}$ beschreibe die Kante von a nach \tilde{a} in G . Die Anzahl der von a ausgehenden Kanten werde mit $d(a)$ („degree“) bezeichnet. Im Folgenden wird ein zeitkontinuierlicher Quantum-Walk beschrieben.

Die Wahrscheinlichkeit von einem Knoten zu einem damit verbundenen Knoten zu gelangen in einer Zeit ϵ sei $\gamma\epsilon$. Bei N Knoten kann das noch immer klassische Problem mit einer $N \times N$ -Matrix K beschrieben werden über

$$K_{a\tilde{a}} = \begin{cases} \gamma & a \neq \tilde{a}, \quad a\tilde{a} \in G \\ 0 & a \neq \tilde{a}, \quad a\tilde{a} \notin G \\ -d(a)\gamma & a = \tilde{a}. \end{cases} \quad (3.15)$$

Sei $p_a(t)$ die Wahrscheinlichkeit, zur Zeit t am Knoten a zu sein, dann gilt

$$\frac{\partial p_a(t)}{\partial t} = \sum_{\tilde{a}} K_{a\tilde{a}} p_{\tilde{a}}(t). \quad (3.16)$$

Gemäß [Chil02@] kann diese Dynamik übertragen werden auf ein quantenbasiertes System, dessen Hilbertraum groß genug ist, um den Knoten entsprechende Basisvektoren zu enthalten. Der dabei benötigte Hamiltonoperator muss so beschaffen sein, dass

$$\langle a|H|\tilde{a}\rangle = K_{a\tilde{a}} \quad (3.17)$$

gilt. Zur Vereinfachung können die Diagonalelemente auf Null gesetzt werden.

Konkrete Beschreibung von H

Der Hamiltonoperator kann erst mit Hilfe eines kleinen Tricks beschrieben werden. Jede Kante soll eine Farbe erhalten, von jedem Knoten aus dürfen keine zwei Kanten derselben Farbe ausgehen. Ist Δ die in dem Graphen maximale Zahl von einem Knoten ausgehender Kanten, kann gezeigt werden ([Chil02@]), dass höchstens $\Delta + 1$ Farben für eine konsistente Farbgebung der Kanten nötig sind.

Sei nun $n = \lceil \log_2 N \rceil$, so werden n Bits für die Knoten (Bitstring) von G und $2n$ Bits für die Knoten von G_k benötigt. Als Black-Box betrachtet wird zu einem gegebenen Knoten und einer Kantenfarbe derjenige Knoten ausgegeben, der über die entsprechende Kante mit dem Knoten verbunden ist. Gibt es keinen solchen Knoten, muss eine Dummy-Ausgabe a_D erstellt werden (ein freier Bitstring). Das ganze wird in einem zweiten Bitstring b festgehalten. Also: $v_c(a) = \tilde{a}$ mit der Farbe c . Es gilt $v_c(v_c(a)) = a$. So lässt sich mit a, b (gewisse Bitstrings jeweils der Länge $2n$) für jede Farbe c eine Quanten-Black-Box mit der bitweisen Addition modulo 2 „ \oplus “ beschreiben. Um eine Dummy-Ausgabe abzufangen wird ein zusätzliches q-Bit eingeführt.

$$V_c|a, b, r\rangle = |a, b \oplus v_c(a), r \oplus f_c(a)\rangle \quad (3.18)$$

mit

$$f_c(a) = \begin{cases} 0 & v_c(a) \neq a_D \\ 1 & v_c(a) = a_D \end{cases}$$

Der Hamiltonoperator des QA muss folgendes gewährleisten:

$$H|a, 0, 0\rangle = \sum_{c: v_c(a) \in G} |v_c(a), 0, 0\rangle \quad (3.19)$$

Mit dem Hermiteschen Operator

$$T|a, b, 0\rangle = |b, a, 0\rangle, \quad T|a, b, 1\rangle = 0$$

entsteht der für das Graphenproblem gesuchte Hamilton-Operator

$$H = \sum_c V_c^\dagger T V_c \quad (3.20)$$

Die Diskretisierung gemäß Abschnitt 3.2 von H kann nun im Wesentlichen mit den Hilfsmitteln der „Unitären Transformation“ und der „Addition“ durchgeführt werden. Dazu muss jedoch noch ein zusätzliches q -Bit hinzugefügt werden (vgl. [Chil02@]). Es werden bei einer Problemstellung eines Quantum-Walk mit einem Graphen G und N Knoten insgesamt zur Simulation mit Gates genau

$$2n + 2n + 2 = 4n + 2, \quad n = \lceil \log_2 N \rceil \quad (3.21)$$

q -Bits benötigt. Die Quanten-Black-Box, d.h. die Farbmatrizen, sind permutierte Einheitsmatrizen und so bereits unitär. Es wird T über $e^{-iT\frac{t}{m}}$ mit Hilfe von Phasenverschiebungen $e^{-i\frac{t}{m}}$, Rotationen mit $\cos(\frac{t}{m})$ bzw. $-i\sin(\frac{t}{m})$ und permutierten Einheitsmatrizen erzeugt. Ein Quantenschaltkreis für e^{-iTt} , der zur Simulation von H benötigt wird, ist in ([Chil02@]) zu finden. Ein Beispiel zur Umsetzung findet sich im Kapitel der Simulationsergebnisse. Festzuhalten bleiben als problematischste Technik des Quantum-Walk die stets nach dem selben Schema vorkommenden Rotationen. Die Phasenverschiebungen und die Permutationsmatrizen kosten vergleichsweise keinen Rechenaufwand.

3.4 Zusammenfassung

Die vorhandene Zahl unterschiedlicher QA kann auf vielfältige Weise kategorisiert werden. Im Hinblick auf die im nächsten Kapitel erläuterte Studie zur Simulation dieser Algorithmen mit Hilfe eines stochastischen Algorithmus in Form eines Branch&Bound-Verfahrens ist eine Unterscheidung dahingehend zu tätigen, inwieweit sich die Spalteneinträge der unitären Matrizen im Betrag unterscheiden. Sind viele überwiegend betragsgleiche Spalteneinträge vorhanden - wie es bei der Hadamard-Transformation der Fall ist - wird sich zeigen, dass das für die Simulation nachteilig ist. Die Klassen der Hidden Subgroup-Algorithmen, der Amplitudenverstärkung und der Quantum-(Random)-Walks zählen dazu. Gibt es dagegen, wie bei den Quantum-Walks, überwiegend sich im Betrag stark unterscheidende Spalteneinträge, kann das für die Simulation von Vorteil sein. Ein wichtiges Ergebnis ist weiterhin, dass sich in der Regel die zeitkontinuierliche und die zeitdiskrete Darstellung von QA äquivalent ineinander transformieren lassen.

QA sind zunächst einmal nicht mit klassischen stochastischen Algorithmen vergleichbar. Denn die dabei betrachteten Zufallsvariablen hängen jeweils von allen Realisierungen ihrer Vorgänger-Zufallsvariablen ab und nicht wie per Definition gefordert von einer Einzigen. Die Vergleichbarkeit kann aber durch einen einkalkulierten Informationsverlust in jedem Algorithmuschritt erreicht und soll nun algorithmisch umgesetzt werden.

KAPITEL 4

Simulation mit einem Branch&Bound-Verfahren

In Abschnitt 3.1.3 wurde bereits eine stochastische Interpretation eines Quantenalgorithmus dargestellt. Dabei wurde unter Beibehaltung der vollen Komplexität und ohne Informationsverlust mit (3.3) das exakte Ergebnis bestimmt. Die serielle Anwendung der Ein-Bit-Operationen wird zwar als Folge von Zufallsvariablen beschrieben, jedoch ist sie kein klassischer stochastischer Algorithmus. Der Vorgang wird nunmehr mit Hilfe von Gleichung (3.9) zu einem stochastischen Algorithmus geändert. Damit ist ein Informationsverlust nach (3.10) verbunden. Der Vorteil davon ist aber die Reduktion der Komplexität der Berechnung.

4.1 Algorithmusidee

Seien in einem Quantenschaltkreis ein Startvektor e_j und die unitäre Matrix

$$M = R_1 \cdot \dots \cdot R_s$$

gemäß (2.12) gegeben. Bei jeder Anwendung einer Bit-m-Operation R_k auf ein Multi-q-Bit wird eine Komponente des Vektors aufgespalten in die Linearkombination zweier Basisvektoren. Die Matrix-mal-Vektor Kombination bildet einen (skalierten) Basisvektor e_j auf die jeweilige Spalte $r_{\cdot j}^i$ der aktuellen Matrix R_i ab. Dies wird für alle Komponenten des Vektors durchgeführt. Sukzessive folgen die weiteren Operationen. Das Ganze kann - startet man von einem Startvektor aus - als Binärbaum, der die Tiefe s entsprechend der Anzahl an Bit-m-Operationen besitzt, dargestellt werden. Eine Darstellung zweier Verzweigungsschritte ausgehend vom Basisvektor e_j zeigt Abbildung 4.1.

Dabei entstehen 2^s Blätter. Jedes Blatt liefert einen (kleinen) Beitrag zum Gesamtergebnis. Es gibt Blätter, die einen größeren Beitrag liefern als Andere. Werden deshalb zunächst solche Blätter berechnet und aufsummiert, so wird im günstigen Fall ein approximativ gutes Ergebnis mit wenigen Rechenoperationen erzielbar. Ist die erwartete Änderung jeder Ergebnisvektorkomponente so gering, dass sich keine signifikanten Änderungen des Ergebnisses ergeben sollten, wird abgebrochen. Der Ausdruck signifikant lässt sich folgendermaßen erklären: Oft sind Quantenalgorithmen so konstruiert, dass auf bestimmten Zuständen eine größere Wahrscheinlichkeit liegt als auf den Anderen. Ist nun etwa die erwartete Änderungsmöglichkeit einer Komponente so klein, dass sich an der Größenordnung der Wahrscheinlichkeitsverteilung im Vergleich zu den anderen Komponenten nichts mehr ändert, kann das Verfahren abgebrochen werden.

Deshalb ist das Ziel des Verfahrens, diejenigen Blätter zu bestimmen, deren Beiträge zum Gesamtergebnis am größten sind. Für den Algorithmus wird eine Anfangsheuristik (1) benötigt. Zunächst soll von der Wurzel aus entlang des Pfades zu einem der Blätter gegangen werden,

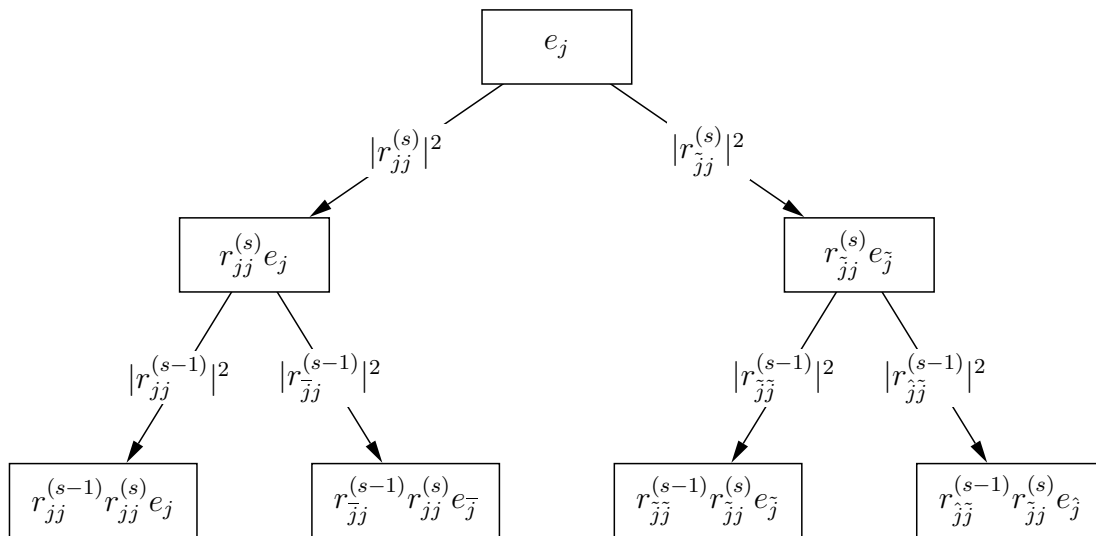


Bild 4.1: Interpretation des Verzweigungsprozesses als Binärbaum

bei dem sich an jedem Knoten gemäß der Kantenwahrscheinlichkeiten (siehe Abbildung 4.1) verzweigt wird. Dabei gibt es zwei Möglichkeiten. Die Erste ist, sich stets in Richtung der größeren Kantenwahrscheinlichkeit und bei Gleichheit zufällig zu bewegen, die Zweite wählt immer per zufälliger Entscheidung die Richtung aus. Die Wahrscheinlichkeiten für die einzelnen Richtungen entsprechen dann den jeweiligen Kantenwahrscheinlichkeiten (z.B. entsteht vom Wurzelknoten aus die Aufteilung $|r_{jj}^{(s)}|^2 \wedge |r_{\tilde{j}\tilde{j}}^{(s)}|^2$). Die Interpretation als stochastischer Algorithmus ist in beiden Fällen gleich: Die Verzweigung wird mit Hilfe von (3.9) modelliert und hat den in (3.10) quantifizierten Informationsverlust zur Folge.

Mit dem erreichten Blatt existiert dann eine erste Unterschranke der Pfadbewertungen bei der Suche. Es wird dann entweder in Richtung der Wurzel gegangen oder in Richtung eines der Blätter (2). Bei jedem Schritt nach oben wird überprüft, ob in die andere Abwärtsrichtung für die Bewertung wenigstens der Wert der Unterschranke erreicht werden kann. Ist dies der Fall wird abwärts gegangen, solange der Unterschrankenwert erzielbar ist. Dabei wird wieder Information gewonnen. Allerdings ist er erst dann bemerkbar, wenn ein weiteres Blatt erreicht wird. Geschieht das, wird die Unterschranke neu gesetzt. Überschreitet man dagegen die Wurzel des Baums, ist die Suche nach dem Blatt, das den aktuell größten Beitrag liefert, abgeschlossen. Dieser Pfad wird markiert und für die weitere Untersuchung ausgeschlossen und die Suche wird auf dem relaxierten Problem neu gestartet (4). So wird weiter verfahren, bis ein festgelegtes Abbruchkriterium (3) erfüllt wird. Das können die Anzahl der Rechenoperationen (z.B. Multiplikationen), der bereits berechnete Beitrag zum Gesamtergebnis usw. sein.

Bemerkung. Wird das als stochastischer Algorithmus gemäß (3.9) formuliert, geschieht folgendes: Der gewählte Startvektor ist die Realisierung der ersten Zufallsvariablen X_0 des Algorithmus. Über das nächste Gate ist dann eine neue Verteilung der nächsten Zufallsvariablen X_1 bestimmt. Dann wird aus den zwei sich ergebenden Möglichkeiten wiederum eine zufällig ausgewählt und es

erfolgt die nächste Realisierung. Das setzt sich bis zum letzten Algorithmusschritt so weiter fort.

4.2 Umsetzung des Verfahrens

Die zur Reduzierung der Komplexität eines Quantenalgorithmus entstandene Algorithmusidee kann in Anlehnung an ein Branch&Bound-Verfahren (nach [Borg01]) gelöst werden.

Es soll nun die Algorithmusidee konkret umgesetzt werden.

(1) Initialisierung:

Finde mit primaler Heuristik eine zulässige Lösung mit der Bewertung U , die als Unterschranke des Problems dient. Wähle dazu (zufällig gemäß der Verteilung der Matrixspalte) den Nachfolger vom aktuellen Knoten $x^{(q)}$ der Tiefe q aus und setze $u(x^{(s)}) = \prod_{i=1}^{(s)} r_{k_i l_i}^i$. Die Unterschranke ist dann $U = |u(x^{(s)})|^2$. Setze den Ergebnisvektor $v \in \mathbb{C}^{2^n}$ mit $v = u(x^{(s)})e_l$, mit e_l als den zu dem erreichten Blatt gehörenden Basisvektor.

(2) Duale Heuristik:

Gehe den aktuellen Pfad rückwärts in Richtung Wurzelknoten. Bei jedem Knoten $x^{(q)}$ der Tiefe q , dessen Bewertung größer ist als die Summe aus U und der Bewertung des Knotens $x^{(q+1)}$, von dem aus $x^{(q)}$ erreicht wurde,

$$U + |u(x^{(q+1)})|^2 \leq |u(x^{(q)})|^2,$$

gehe den Alternativpfad, sofern noch nicht getan, zum nächsten Knoten. Wurde der Alternativpfad bereits beschritten, muss weiter in Richtung Wurzelknoten entlang gegangen werden. Existiert kein solcher Knoten x und wird somit der Wurzelknoten erreicht \rightarrow Gehe zu (4). Ansonsten folge dem Pfad solange nach unten (mit primaler Heuristik), bis ein Knoten $\hat{x}^{(p)}$ erreicht wird, dessen Bewertung kleiner als U ist ($U > |u(\hat{x}^{(p)})|^2$). \rightarrow Gehe zu (3). Sollte ein Blatt erreicht werden, muss der Ergebnisvektor aktualisiert werden zu

$$v = v + e_l \cdot |u(\hat{x}^{(s)})|^2,$$

wobei e_l der hierbei erreichte Basisvektor ist. Falls $|u(\hat{x}^{(s)})|^2 > U$ ist, setze $U = |u(\hat{x}^{(s)})|^2$ neu. Setze die Suche im Unterbaum fort, bis er abgeschlossen ist.

(3) Abbruchkriterien:

Zu viele Rechenoperationen wurden bereits durchgeführt \rightarrow ENDE

Eine ausreichende Näherung an das Ergebnis ist erreicht,

$\|v\|_{\mathbb{C}}$ genügend groß \rightarrow ENDE

Ansonsten \rightarrow Gehe zu (2).

(4) Nach dem Finden des Blattes mit dem nächstgrößten Beitrag zum Ergebnisvektor muss U neu eingestellt werden. Setze hierzu U auf die Bewertung des beitragsbezogen gleich- oder nächstgrößten bereits berechneten Blattes. Gibt es kein weiteres Blatt, so suche den Knoten,

der zwei nicht-berechnete Nachfolgeknoten besitzt, mit der nächstgrößten Bewertung und verfolge den Pfad mit der maximaler Nachfolger-Heuristik bis zu einem Blatt und setze U neu. Wird kein weiteres Blatt erreicht \rightarrow ENDE. Setze als aktuellen Knoten das eben erreichte Blatt und \rightarrow Gehe zu (2).

Auf diese Weise erzielen die Abbruchkriterien in (3) auf natürliche Weise eine größere Bedeutung und dienen als Feineinstellung bei einem Ablauf des Algorithmus. Allerdings stehen die beiden Kriterien in gegenseitiger Konkurrenz: Je mehr erklärt werden soll, d.h. je mehr Pfade und Blätter berechnet werden, desto mehr Rechenoperationen sind auch auszuführen. Andererseits werden Pfade durch eine Beschränkung der Anzahl an Rechenoperationen ausgeschlossen. Letztendlich werden durch das Einführen von Schritt (4) auf rekursive Weise neue Blattbewertungen in den Ergebnisvektor aufgenommen. Fließen alle Blattbewertungen ein, so gilt

$$\|v\|_{\mathbb{C}} = 1,$$

was sich aus der Struktur gemäß Abbildung 4.1 leicht ableiten lässt. Wird das Verfahren abgebrochen, kann untersucht werden, inwieweit das Ergebnis ausreichend gut ist. Dies zeigt folgender Abschnitt.

4.3 Güte des Verfahrens

Das Verfahren der allmählichen Aufsummierung der Wahrscheinlichkeiten geht einher mit der Konstruktion eines zugehörigen Lösungsvektors. Wird das Verfahren durch das Erfüllen eines Abbruchkriteriums gestoppt, so muss geklärt werden, wie exakt der konstruierte Ergebnisvektor dem tatsächlichen Ergebnisvektor angenähert worden ist. Das soll aber nicht genau bestimmt werden, da ansonsten alle Wahrscheinlichkeiten und damit der gesamte binäre Baum berechnet werden müsste. Dies führt zu einem exponentiellen Aufwand, der ja gerade vermieden werden sollte. Eine andere Möglichkeit wäre, eine Abschätzung zu finden, um wie viel sich eine jede Komponente des konstruierten Ergebnisvektors nach dem Stopp des Verfahrens noch in beliebiger Richtung verändern kann.

4.3.1 Problemdefinition

Das Verfahren berechnet näherungsweise die Anwendung verschiedener Bit-m-Operationen R_i , $i = 1, \dots, s$ auf einen festgelegten Basisvektor.

$$R_1 \cdot \dots \cdot R_s \cdot e_j, \quad R_i \in \mathbb{C}^{n,n}, \text{ unitär} \quad (4.1)$$

Während des Verfahrens werden Pfade durch den aus dem Matrix-mal-Vektor Produkt in (4.1) entstehenden Binärbaum der Tiefe s berechnet und aufgesammelt. Ein solcher Pfad, der geschrieben werden kann als

$$s_j = \left(\prod_{i=1}^s r^{(i)} \right) e_j, \quad r^{(i)} \in \mathbb{C}, \quad 0 \leq |r^{(i)}|^2 \leq 1 \quad (4.2)$$

liefert einen Beitrag zur j -ten Komponente des Ergebnisvektors. Sei nun nach dem Stopp des Verfahrens

$$k = \sum_{j=1}^n k_j e_j = \begin{pmatrix} k_1 \\ \vdots \\ k_n \end{pmatrix}, \quad k_j \in \mathbb{C} \quad \forall j = 1, \dots, n \quad (4.3)$$

der konstruierte Ergebnisvektor. Eine Komponente k_j von k ist durch das Aufsummieren aller m_j berechneten und auf den Basisvektor e_j führenden Blattwerte $s_{ji} \in \mathbb{C}$, $i = 1, \dots, m_j$ entstanden:

$$k_j = \sum_{i=1}^{m_j} s_{ji} \quad (4.4)$$

Die nach Abbruch des Verfahrens noch nicht berechneten Beiträge $a_{ji} \in \mathbb{C}$, $i = 1, \dots, n_j$ zu den einzelnen Komponenten j des Ergebnisvektors liefern bei Betrachtung als Wahrscheinlichkeiten zusammen mit den bereits berechneten Beiträgen die Gesamtwahrscheinlichkeit:

$$\underbrace{\sum_{j=1}^n \sum_{i=1}^{m_j} s_{ji} \bar{s}_{ji}}_{\stackrel{\text{def}}{=} (1-p)} + \underbrace{\sum_{j=1}^n \sum_{i=1}^{n_j} a_{ji} \bar{a}_{ji}}_{\stackrel{\text{def}}{=} p} = 1 \quad (4.5)$$

Das bedeutet, dass die Summe der quadrierten Beträge der berechneten Blätter und die Summe der quadrierten Beträge der noch nicht berechneten Blätter, also $(1-p) + p$, die Gesamtwahrscheinlichkeit 1 ergibt. Denn betrachtet man den Binärbaum von oben, so kann das über Induktion nach der Tiefe q des Baumes gezeigt werden. Es ist dabei nicht von Belang, ob ein Pfad des Baumes und damit ein Blatt bereits berechnet wurde oder nicht.

Beweis. Induktion über q .

$$q = 0: \quad 1 \cdot e_j \rightarrow |1|^2 = 1 \quad \checkmark$$

$$q = 1: \quad r_{jj}^{(m)} e_j + r_{\tilde{j}j}^{(m)} e_{\tilde{j}} \xrightarrow{R_i \text{ unitär}} |r_{jj}^{(m)}|^2 + |r_{\tilde{j}j}^{(m)}|^2 = 1$$

$q \rightarrow q+1$: Aussage sei für q richtig.

Es gibt einen aktuellen Lösungsvektor $r_1 e_1 + \dots + r_n e_n$ mit $|r_1|^2 + \dots + |r_n|^2 = 1$.

Aufteilung: $r_j e_j \rightarrow r_j r_j^{(q+1)} e_j + r_j r_k^{(q+1)} e_k = r_j (r_j^{(q+1)} e_j + r_k^{(q+1)} e_k)$.

Wegen $|r_j^{(q+1)}|^2 + |r_k^{(q+1)}|^2 = 1$ gilt

$$\begin{aligned} |r_j r_j^{(q+1)}|^2 + |r_j r_k^{(q+1)}|^2 &= r_j r_j^{(q+1)} \bar{r}_j \bar{r}_j^{(q+1)} + r_j r_k^{(q+1)} \bar{r}_j \bar{r}_j^{(q+1)} \\ &= r_j \bar{r}_j (r_j^{(q+1)} \bar{r}_j^{(q+1)} + r_k^{(q+1)} \bar{r}_k^{(q+1)}) = |r_j|^2 \end{aligned}$$

Das gilt für jede Basiskomponente und damit bleibt insgesamt die Länge 1 erhalten. □

Die Summe der Wahrscheinlichkeiten, die sich mit den bereits bestimmten s_{ji} gruppiert nach einem Basisvektor e_j aufaddieren lässt, entspricht aber nicht der Wahrscheinlichkeit, die sich aus der einzelnen Komponente des errechneten Ergebnisvektors ergibt.

Satz 4.1 (Aufsummieren von Wahrscheinlichkeiten)

$$\left| \sum_{i=1}^{m_j} s_{ji} \right|^2 \stackrel{i.A.}{\neq} \sum_{i=1}^{m_j} |s_{ji}|^2 \quad (4.6)$$

Beweis.

$$\begin{pmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{pmatrix} \cdot \begin{pmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{pmatrix} \cdot e_1 = e_1 \quad \text{mit } e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Die beiden Blätter für e_1 besitzen die Koeffizienten $s_{11} = 0.64$ und $s_{12} = 0.36$.

$$\left| \sum_{i=1}^2 s_{1i} \right|^2 = (0.64 + 0.36)^2 = 1 \neq 0.5392 = 0.64^2 + 0.36^2 = \sum_{i=1}^2 |s_{1i}|^2$$

Es gilt:

$$\begin{aligned} \left| \sum_{i=1}^{m_j} s_{ji} \right|^2 &= (s_{j1} + \dots + s_{jm_j})(\bar{s}_{j1} + \dots + \bar{s}_{jm_j}) \\ &= \sum_{i=1}^{m_j} |s_{ji}|^2 + \sum_{i=1}^{m_j} \sum_{\substack{k=1 \\ k \neq i}}^{m_j} s_{ji} \bar{s}_{jk} \\ &= \sum_{i=1}^{m_j} |s_{ji}|^2 + \underbrace{2 \operatorname{Re} \left(\sum_{i=1}^{m_j} \sum_{\substack{k=2 \\ k > i}}^{m_j} s_{ji} \bar{s}_{jk} \right)}_{\text{nicht zwingend} = 0} \end{aligned}$$

□

Korollar 4.2

$$\sum_{j=1}^n \left| \sum_{i=1}^{m_j} s_{ji} \right|^2 \stackrel{i.A.}{\neq} \sum_{j=1}^n \sum_{i=1}^{m_j} |s_{ji}|^2 \quad (4.7)$$

Jedoch führt die Eigenschaft, dass alle Matrizen R_i unitär sind, dazu, dass von einem bereits konstruierten Ergebnisvektor aus durch das Berechnen aller weiteren Beiträge, d.h. letztendlich durch das Berechnen des gesamten Baumes, sich die Wahrscheinlichkeiten wieder zu 1 addieren müssen.

Satz 4.3 (Erhaltung der Länge)

$$\sum_{j=1}^n \left| k_j + \sum_{i=1}^{n_j} a_{ji} \right|^2 = 1 \quad (4.8)$$

Beweis. Die Multiplikation in (4.1) bildet einen Basisvektor der Länge 1 wieder auf einen Vektor der Länge 1 ab. Das Berechnen aller Beiträge zum Ergebnisvektor liefert das exakte Ergebnis der Multiplikation und damit wieder einen Vektor der Länge 1. □

Ein wichtiges Ergebnis im weiteren Verlauf liefert folgende Rechnung, die sich aus dem letzten Satz ergibt:

$$\begin{aligned}
& \sum_{j=1}^n \left| k_j + \sum_{i=1}^{n_j} a_{ji} \right|^2 \\
&= \sum_{j=1}^n \left| \sum_{i=1}^{m_j} s_{ji} + \sum_{i=1}^{n_j} a_{ji} \right|^2 \\
&= \sum_{j=1}^n \left(\left| \sum_{i=1}^{m_j} s_{ji} \right|^2 + \left| \sum_{i=1}^{n_j} a_{ji} \right|^2 + \sum_{i=1}^{m_j} s_{ji} \sum_{i=1}^{n_j} \bar{a}_{ji} + \sum_{i=1}^{n_j} a_{ji} \sum_{i=1}^{m_j} \bar{s}_{ji} \right) \\
&= \sum_{j=1}^n \left(\underbrace{2\operatorname{Re} \left(\sum_{i=1}^{m_j} \sum_{\substack{k=2 \\ k>i}}^{m_j} s_{ji} \bar{s}_{jk} \right) + 2\operatorname{Re} \left(\sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} a_{ji} \bar{a}_{jk} \right) + 2\operatorname{Re} \left(\sum_{i=1}^{m_j} s_{ji} \sum_{i=1}^{n_j} \bar{a}_{ji} \right)}_{=0} \right) \\
&\quad + \underbrace{\sum_{j=1}^n \left(\sum_{i=1}^{m_j} |s_{ji}|^2 + \sum_{i=1}^{n_j} |a_{ji}|^2 \right)}_{\stackrel{(4.5)}{=} 1} \\
&\stackrel{(4.8)}{=} 1
\end{aligned} \quad (4.9)$$

Interessant ist nun, eine Abschätzung zu finden, die angibt, um wie viel sich nach Abbruch des Verfahrens die einzelnen Komponenten des Ergebnisvektors noch ändern könnten. Das zeigt der folgende Abschnitt.

4.3.2 Abschätzung der Änderungsmöglichkeiten

Bei der Abschätzung der noch möglichen Änderungen des konstruierten Ergebnisvektors sind drei Fälle zu unterscheiden:

- der günstigste Fall („best case“), bei dem sich keine schwerwiegenden Änderungen mehr ergeben
- der erwartete Fall („average case“), ein Erwartungswert für die Problemstellung
- der schlechtest mögliche Fall („worst case“), der zeigt, dass das Verfahren ein trügerisches Zwischenergebnis liefern kann.

Am schwierigsten zu behandeln ist der average case. Deshalb wird er als Letzter in diesem Abschnitt durchgeführt. Die Formulierung der Änderungsmöglichkeit des konstruierten Ergebnisvektors bedeutet nichts Anderes, als den Ausdruck

$$\left| \sum_{j=1}^n \sum_{i=1}^{n_j} a_{ji} \right|^2 \quad (4.10)$$

bzw. gruppiert nach den einzelnen Komponenten j

$$\sum_{j=1}^n \left| \sum_{i=1}^{n_j} a_{ji} \right|^2 \quad (4.11)$$

unter gewissen Nebenbedingungen, die aus dem Problem in (4.1) heraus entstehen, abzuschätzen. Die Wurzel aus (4.10) bzw. (4.11) ergibt dann den Betrag der Änderung.

4.3.3 „best case“-Abschätzung

Optimal ist es, falls zum Zeitpunkt des Abbruchs des Verfahrens der konstruierte Ergebnisvektor gleich dem tatsächlichen Ergebnis ist. Das bedeutet, die mögliche Änderung ergibt 0

$$\sum_{j=1}^n \left| \sum_{i=1}^{n_j} a_{ji} \right|^2 = 0 \quad (4.12)$$

und somit

$$(4.12) \Leftrightarrow 2\operatorname{Re}\left(\sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} a_{ji} \bar{a}_{jk}\right) = -p, \quad 0 \leq p \leq 1. \quad (4.13)$$

Denn der Ausdruck in (4.11) lässt sich schreiben als

$$\begin{aligned} (4.11) &\stackrel{(4.9)}{=} \sum_{j=1}^n \left(\sum_{i=1}^{n_j} |a_{ji}|^2 + 2\operatorname{Re}\left(\sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} a_{ji} \bar{a}_{jk}\right) \right) \\ &= p + 2\operatorname{Re}\left(\sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} a_{ji} \bar{a}_{jk}\right) \end{aligned} \quad (4.14)$$

(4.12) gilt also genau dann, wenn der zweite Summand in (4.14) zu $-p$ wird.

Bemerkung. Zwei Spezialfälle sind bei $p = 0$ bzw. $p = 1$ gegeben. Ist nämlich $p = 0$, so ist der gesamte Baum bereits berechnet und der zweite Summand verschwindet, weil es keine a_{ji} mehr zu bestimmen gibt. Ebenso verschwindet er, falls $p = 1$ ist und damit noch der gesamte Baum berechnet werden muss (vgl. (4.5) bzw. (4.9)). Diese Bestimmung ist nicht überraschend. Allerdings sind die Interpretation der Gleichung (4.13) und die Schreibweise der Gleichung (4.11) gemäß (4.14) für die weitere Untersuchung von Bedeutung.

4.3.4 „worst case“-Abschätzung

Die zweite interessante Untersuchung zeigt, inwieweit der Ergebnisvektor im ungünstigsten Fall noch von dem aktuellen Ergebnis abzuweichen imstande ist. Es soll der maximale Betrag der Abweichungen aufsummiert über alle Komponenten bestimmt werden. Um eine feinere Abschätzung zu finden, können die einzelnen noch nicht berechneten Beiträge auf die einzelnen Komponenten zugewiesen werden. Dazu ist folgendes Problem zu lösen:

$$\begin{aligned} \text{Problemstellung. } \max \quad \text{ZF} &= \sum_{j=1}^n \left| \sum_{i=1}^{n_j} a_{ji} \right|^2 \\ \text{unter (1)} \quad &(4.5) \\ &(2) \quad (4.8) \\ &(3) \quad \sum_{j=1}^n |k_j|^2 = K \end{aligned}$$

Die Nebenbedingung (2) lässt sich mit $a_j = (a_{j1}, \dots, a_{jn_j})^T$ umformen zu

$$\begin{aligned} 1 &= \sum_{j=1}^n \left| k_j + \sum_{i=1}^{n_j} a_{ji} \right|^2 \\ &= \sum_{j=1}^n \left[\left(k_j + \sum_{i=1}^{n_j} a_{ji} \right) \left(\bar{k}_j + \sum_{i=1}^{n_j} \bar{a}_{ji} \right) \right] \\ &= \sum_{j=1}^n \left(|k_j|^2 + \left| \sum_{i=1}^{n_j} a_{ji} \right|^2 + 2 \operatorname{Re} \left(k_j \sum_{i=1}^{n_j} \bar{a}_{ji} \right) \right) \\ &\stackrel{(3)}{=} K + \text{ZF} + 2 \sum_{j=1}^n \operatorname{Re} \left(k_j \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^T \bar{a}_j \right) \\ &\Leftrightarrow \text{ZF} = 1 - K - 2 \underbrace{\sum_{j=1}^n \operatorname{Re} \left(k_j \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^T \bar{a}_j \right)}_{\text{zu minimieren}} \end{aligned}$$

Es ergibt sich so ein neues Problem.

$$\text{Problemstellung 2. } \min \quad ZF_2 = \sum_{j=1}^n \operatorname{Re}(k_j \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^T \bar{a}_j)$$

unter (1) (4.5)

$$ZF_2 \stackrel{\operatorname{Re}(z) \geq -|z|}{\geq} \sum_{j=1}^n -|k_j \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^T \bar{a}_j|$$

$$\stackrel{|z_1 z_2| = |z_1| |z_2|}{=} - \sum_{j=1}^n |k_j| \cdot \left| \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^T \bar{a}_j \right|$$

Da nun alle $|k_j| \geq 0$ sind, lässt sich Problemstellung 2 abschätzen mit

$$\text{Problemstellung 3. } \max \quad ZF_3 = \sum_{j=1}^n \left| \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^T \bar{a}_j \right|$$

unter (1) (4.5)

$$\sum_{j=1}^n \left| \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^T \bar{a}_j \right| \leq \sum_{j=1}^n \left\| \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \right\| \cdot \|\bar{a}_j\| = \sum_{j=1}^n \sqrt{n_j} \cdot \|a_j\|$$

Die Ungleichung wird scharf genau dann wenn alle Komponenten in a_j gleich sind, $\bar{a}_j = \hat{a}_j$
 $\bar{a}_{ji} = \bar{a}_{jk} \forall i, k$.

$$\sum_{j=1}^n \left| \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^T \bar{a}_j \right| = \sum_{j=1}^n \sqrt{n_j \hat{a}_j \cdot n_j \bar{\hat{a}}_j} = \sum_{j=1}^n \sqrt{n_j} \sqrt{n_j} |\hat{a}_j| = \sum_{j=1}^n \sqrt{n_j} \cdot \|\hat{a}_j\|$$

$$\text{Problemstellung 4. } \max \quad ZF_4 = \sum_{j=1}^n \sqrt{n_j} \cdot \|\hat{a}_j\| = \sum_{j=1}^n n_j |\hat{a}_j|$$

unter (1) (4.5)

Unter diesen Umständen lautet die Nebenbedingung (1) dann

$$\sum_{j=1}^n n_j |\hat{a}_j|^2 = p$$

Setze nun $|\hat{a}_j| = \sqrt{\frac{p}{n_j n}}$. Dann wird die Nebenbedingung (1) erfüllt und es ergibt sich das Maximum für ZF_4 . Für das Maximum der Ausgangsfunktion ergibt sich damit

$$\begin{aligned} \sum_{j=1}^n \left(\sum_{i=1}^{n_j} a_{ji} \sum_{i=1}^{n_j} a_{ji} \right) &= \sum_{j=1}^n n_j^2 |\hat{a}_j|^2 \\ &= \sum_{j=1}^n n_j^2 \left(\sqrt{\frac{p}{n_j n}} \right)^2 \\ &= \frac{p}{n} \sum_{j=1}^n n_j \end{aligned}$$

Der Betrag der maximalen Änderungsmöglichkeit für den Ergebnisvektor ergibt sich als

$$\frac{1}{\sqrt{n}} \cdot \sqrt{p} \cdot \sqrt{\sum_{j=1}^n n_j} \quad (4.15)$$

4.3.5 „average case“-Abschätzung

Bei der average case-Abschätzung sollen ein Erwartungswert und eine Varianz für (4.11) bestimmt werden. So kann eine Aussage dahingehend getroffen werden, in welchem Intervall sich zu einer festgelegten Prozentzahl der Fälle erwartungsgemäß die Änderungsmöglichkeiten der Komponenten des konstruierten Ergebnisvektors bewegen.

Allgemeine Abschätzung

Da a-priori nichts weiter über die a_{ji} gesagt werden kann, sei angenommen, dass die a_{ji} unabhängig identisch verteilt sind. Dabei sollen in Polarkoordinatendarstellung die Winkel als von den Beträgen unabhängig angenommen werden.

$$\begin{aligned} E\left[\sum_{j=1}^n \left| \sum_{i=1}^{n_j} a_{ji} \right|^2\right] &= E\left[p + 2\operatorname{Re}\left(\sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} a_{ji} \bar{a}_{jk}\right)\right] \\ &= E[p] + E\left[2\operatorname{Re}\left(\sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} a_{ji} \bar{a}_{jk}\right)\right] \\ &= p + 2 \sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} E[\operatorname{Re}(a_{ji} \bar{a}_{jk})] \end{aligned}$$

$$\begin{aligned}
&= p + 2 \sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} E[\operatorname{Re}(|a_{ji}| e^{i(\varphi_{a_{ji}})} \cdot |\bar{a}_{jk}| e^{-i(\varphi_{a_{jk}})})] \\
&= p + 2 \sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} E[|a_{ji}| \cdot |a_{jk}|] E[\cos(\varphi_{a_{ji}} - \varphi_{a_{jk}})]
\end{aligned} \tag{4.16}$$

Sind alle a_{ji} unabhängig identisch verteilt, dann ist die Differenz X zweier Winkel von a_{ji} bzw. a_{jk} „symmetrisch“ mit Erwartungswert 0 verteilt. Die Bildung des Realteils dieser entstandenen Zufallsvariablen ändert nichts an dieser Symmetrie und dem Erwartungswert. Dann gilt für den Erwartungswert

$$(4.16) = p + 2 \sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} E[|a_{ji}| \cdot |a_{jk}|] \underbrace{E[\cos(\varphi_{a_{ji}} - \varphi_{a_{jk}})]}_{=0} = p \tag{4.17}$$

Die Varianz lässt sich folgendermaßen abschätzen.

$$\begin{aligned}
V\left[\sum_{j=1}^n \left|\sum_{i=1}^{n_j} a_{ji}\right|^2\right] &= E\left[\left(\sum_{j=1}^n \left|\sum_{i=1}^{n_j} a_{ji}\right|^2 - E\left[\sum_{j=1}^n \left|\sum_{i=1}^{n_j} a_{ji}\right|^2\right]\right)^2\right] \\
&= E\left[\left(p + 2\operatorname{Re}\left(\sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} a_{ji}\bar{a}_{jk}\right) - p\right)^2\right] \\
&= E\left[\left(2\operatorname{Re}\left(\sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} a_{ji}\bar{a}_{jk}\right)\right)^2\right] \\
&= 4E\left[\left(\sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} \operatorname{Re}(|a_{ji}| \cdot |a_{jk}| \cdot e^{i(\varphi_{a_{ji}} - \varphi_{a_{jk}})})\right)^2\right] \\
&= 4E\left[\left(\sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} |a_{ji}| \cdot |a_{jk}| \cdot \cos(\varphi_{a_{ji}} - \varphi_{a_{jk}})\right)^2\right] \\
&= 4V\left[\sum_{j=1}^n \sum_{i=1}^{n_j} \sum_{\substack{k=2 \\ k>i}}^{n_j} |a_{ji}| \cdot |a_{jk}| \cdot \cos(\varphi_{a_{ji}} - \varphi_{a_{jk}})\right]
\end{aligned} \tag{4.18}$$

Das ist aber genau die Varianz der nicht-quadratierten Zufallszahl. Es könnte vorab angenommen werden, dass zumindest im Mittel in der Polardarstellung für die a_{ji}

$$a_{ji} = \sqrt{\frac{p}{n_j n}} e^{i\varphi} \tag{4.19}$$

geschrieben werden kann. Jeder Summand von (4.18) ist deshalb - bei Annahme der Unabhängigkeit der Zufallsvariablen und des konstanten Betrages gemäß Annahme (4.19) - von der Form

$$\begin{aligned} V[|a_{ji}| \cdot |a_{jk}| \cdot \cos(\varphi_{a_{ji}} - \varphi_{a_{jk}})] &\stackrel{(4.19)}{=} \frac{p^2}{n_j^2 n^2} \cdot V[\cos(\varphi_{a_{ji}} - \varphi_{a_{jk}})] \\ &= \frac{p^2}{n_j^2 n^2} \cdot E[\cos(X)^2] \end{aligned}$$

Der Erwartungswert $E[\cos(X)^2]$ liegt zwischen 0 und 1. Jeder Summand besitzt somit den Wert

$$V = \frac{kp^2}{n_j^2 n^2}, \quad 0 \leq k \leq 1$$

Es muss noch geklärt werden, wie viele Summanden es gibt. Für jedes n_j gibt es

$$\binom{n_j}{2}$$

Summanden. Es folgt für die Varianz

$$\begin{aligned} V &= 4k \sum_{j=1}^n \binom{n_j}{2} \frac{p^2}{n_j^2 n^2} \\ &= 2k \cdot \sum_{j=1}^n n_j(n_j - 1) \frac{p^2}{n_j^2 n^2} \\ &\leq \frac{2p^2}{n} \quad \text{wegen } 0 \leq k \leq 1 \end{aligned} \tag{4.20}$$

4.4 Zusammenfassung

Das vorgestellte Branch&Bound-Verfahren ist eine Möglichkeit zur weitest möglichen Reduktion der Komplexität eines QA. Durch zusätzliche Berechnungsschritte wird wieder Komplexität hinzugefügt. Ein Gate entspricht einer Bit-m-Operation und gewährleistet so die Vergleichbarkeit der Komplexität zwischen QA und deren klassischer Umsetzung. Mit einer feinen Steuerung der Abbruchkriterien kann nun gezielt eine festgelegte Komplexitätsordnung für einen QA betrachtet und ausgewertet werden. Damit kann getestet werden, ob eine Reduktion der Komplexität bei der Berechnung auf einem klassischen Rechner möglich erscheint, während gleichzeitig das Ergebnis ausreichend gut an das tatsächliche herankommt.

Quantenalgorithmen nutzen das Phänomen der Superposition aus, um gegenüber klassischen Algorithmen im Vorteil zu sein. Mit dem Branch&Bound-Verfahren wird dieser Vorteil wieder verworfen und es wird stattdessen gehofft, durch das Finden möglichst großer Beiträge zur Gesamtwahrscheinlichkeit einzelner Blätter einen Effizienzgewinn gegenüber klassischen Algorithmen zu erhalten. Denn statt der bezüglich der Bitzahl exponentiellen Anzahl zu bestimmender Blätter berechnen zu müssen, sollte eine nicht-exponentielle Zahl an berechneten Blättern genügen,

um das tatsächliche Ergebnis nahezu zu erhalten. Sind jedoch alle Blattbewertungen gleich, was von einer gleichwahrscheinlichen Aufspaltung in Teilbäume kommt, kann es keine Einsparungen geben. Dies ist bei den Hidden Subgroup Problemen, der Amplitudenverstärkung und den Quantum Random Walks der Fall. Für Quantum-Walk Probleme tritt ein bei Weitem anderer Fall ein. Wenige Blätter liefern einen großen Beitrag zur Gesamtwahrscheinlichkeit. Doch die Simulationsergebnisse im nächsten Kapitel zeigen, dass es trotzdem Probleme mit der Vorhersage des tatsächlichen Ergebnisses gibt.

Während des Branch&Bound-Verfahrens in den Algorithmen von Shor bzw. von Grover liefert jedes Blatt des Baumes, das nicht den Koeffizienten Null hat, den selben sehr kleinen Beitrag zur Gesamtwahrscheinlichkeit. Die Blätter werden zufällig bestimmt, da jede Abzweigung gleichwahrscheinlich ist. Die Unterschiede der Koeffizienten des Ergebnisvektors ergeben sich durch unterschiedliche Vorzeichen der Blattkoeffizienten, die sich beim Aufsummieren gegenseitig auslöschen oder aufblähen können.

Die Abschätzung der Güte soll als Anhaltspunkt verstanden werden. Gerade die Annahmen bei der average-case-Abschätzung sind nicht auf vorhandene Probleme übertragbar, da starke und im durch Branching und Bounding gesteuerten Ablauf des QA unhaltbare Annahmen getroffen wurden. Die Simulationsergebnisse des nächsten Kapitels werden dieses Faktum untermauern.

KAPITEL 5

Simulationsergebnisse

Zur konkreten Umsetzung und Berechnung der ersten beiden Schritte eines Quantenalgorithmus auf einem klassischen Rechner mit dem im letzten Kapitel beschriebenen Branch&Bound-Verfahren wurde das in Abbildung 5.1 zu sehende Programm in JAVA implementiert. Die Abbil-

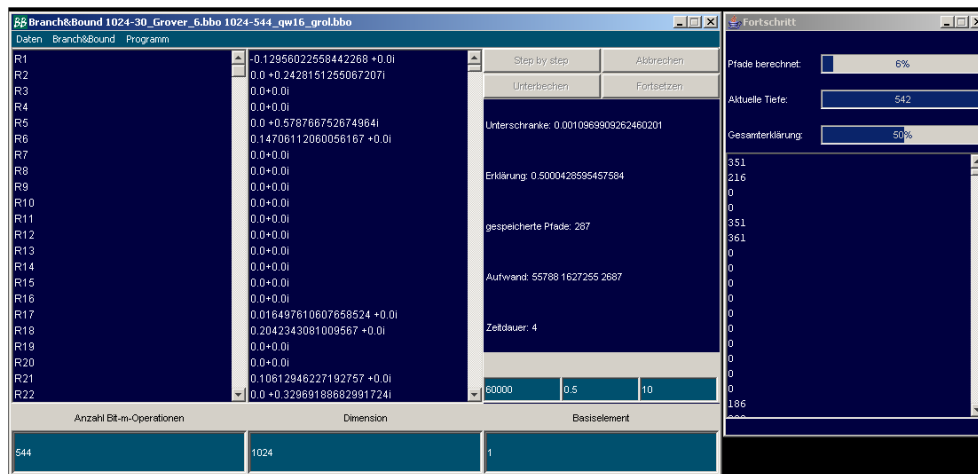


Bild 5.1: Screenshot des Branch&Bound JAVA-Programms

Abbildung 5.1 zeigt die Bearbeitung eines Quantum-Walk-Algorithmus mit 544 Bit-m-Operationen, die im linken Drittel des linken Fensters dargestellt werden. Im mittleren Bereich des Fensters wird der aktuelle Ergebnisvektor (hier mit der Dimension 1024) angezeigt. Der rechte Bereich zeigt verschiedene aktuelle statistische Werte: Von oben nach unten die Unterschranke im Branch&Bound-Verfahren, darunter die mit den bereits berechneten Pfaden des Baumes erreichte Wahrscheinlichkeit, die Anzahl der berechneten Pfade, den für die Berechnung notwendigen Aufwand (Additionen, Multiplikationen und Auswertungen im Sinne des Quantencomputers) und die Zeitdauer für die Berechnung. Die drei Spalten darunter dienen zur Einstellung der Abbruchkriterien des Verfahrens. Hier wurde die Möglichkeit zur Justierung der Auswertungen, der Wahrscheinlichkeit und der Zeitdauer gewählt. Ganz unten rechts ist der Startbasisvektor einstellbar. Das zweite Fenster zeigt weitere Statistiken an. Dazu gehören der Anteil der berechneten Pfade an der Gesamtzahl der Pfade, die Tiefe im Binärbaum und die grafische Darstellung der erreichten Wahrscheinlichkeit. Darunter ist komponentenweise die Anzahl der Blätter mit von Null verschiedenen Beiträgen des Ergebnisvektors angezeigt.

Die für den Ablauf des Programmes benötigten Daten eines QA in Form von Bit-m-Operationen

wurden mit Mathematica 5 erzeugt. In Bild 5.9 ist zu sehen, wie im ersten Schritt ein Gate in Mathematica erstellt wird (hier etwa eine Permutationsmatrix zu $\mathbb{1}_{1024}$). Dabei mussten mindestens für jede Bit- m -Operation das Bit m und für jede Spalte der Matrix zwei komplexe Zahlen in Textdateien abgespeichert werden. Tabelle 5.1 zeigt die jeweils entstandenen Dateigrößen für die einzelnen Aufgaben. Die Probleme wurde so angepasst, dass eine gleiche Anzahl an Bits (10)

Problem	Eckdaten	Dateigröße [MByte]
Suchalgorithmus von Grover	Dimension: 1024, Matrizen: 30	2.9
Quantum-Walk	(i) Dimension: 1024, Matrizen: 544	30.1
	(ii) Dimension: 1024, Matrizen: 1088	60.2

Tafel 5.1: Dateigrößen ausgewählter Datendateien der einzelnen Probleme

zur Simulation benötigt wurden, um eine gute Vergleichbarkeit erzielen zu können.

Beim Ablauf des Programms wurden Ergebnisvektorkomponenten und die Anzahl Pfade gesammelt. Die erreichte Wahrscheinlichkeit wurde schrittweise um 0.05 bzw. 0.025 erhöht und bei Erreichen der Marke die Werte notiert. Aus diesen Daten wurden dann verschiedene Erkenntnisse gewonnen, die nun dargestellt werden sollen.

5.1 Suchalgorithmus von Grover

Die Simulation eines amplitudenverstärkenden QA wurde anhand des Algorithmus von Grover, wie in Abschnitt 3.3.2 beschrieben, durchgeführt. Für das Beispiel wurde zur Konstruktion von Q gemäß Satz 3.9 eine Quanten-Black-Box erstellt, die aus 1024 Werten einen (hier: den Fünften) Wert kennzeichnet. Das bedeutet, die Black-Box U_f liegt als Einheitsmatrix $\mathbb{1}_{1024}$ vor und das fünfte Diagonalelement wird durch den Wert -1 ersetzt. Die Gates des QA bestehen aus genau einer Abfolge folgender 30 Bit- m -Operationen gemäß Abschnitt 3.3.2:

$$\underbrace{H_1 \cdot \dots \cdot H_9 \cdot U_1 \cdot H_9 \cdot \dots \cdot H_1 \cdot U_f}_{Q} \cdot \underbrace{H_{10} \cdot \dots \cdot H_1}_M \quad (5.1)$$

Die H_i stehen für die in Beispiel 3.1.1 beschriebenen Hadamard-Gates, die auf dem i -ten Bit arbeiten. Das Gate $U_1 := -H_{10} \cdot U_0 \cdot H_{10}$ ist eine zu einer Bit-10-Operation zusammengefasste Matrix, wobei U_0 wie in Abschnitt 3.3.2 definiert wird.

Der Startzustand ist der erste Basisvektor e_1 . Das entspricht beispielsweise einer Suche eines bestimmten Datenelements in einer Datenbank. Für die Simulation genügt für die Iterationszahl aus Satz 3.9 bereits die Wahl $m = 1$ und die dadurch in (5.1) beschriebene Sequenz von

Ein-Bit-Operationen, um das Verhalten des Branch&Bound-Verfahrens eindeutig zu zeigen. Nach Durchführung des QA Q^1Me_1 ist die Wahrscheinlichkeit, den fünften Basisvektor bei einer Messung zu erhalten, etwas erhöht. Bei einer Wahl der Iterationszahl von $m = \lfloor \frac{\pi}{4} \sqrt{1024} \rfloor = 25$ läge eine große Wahrscheinlichkeit auf der fünften Komponente (vgl. [Schä02]). Allerdings entstünde daraus ein Problem mit 510 Matrizen, davon 460 Hadamard-Gates, die dann zu 2^{460} Blättern mit dem Blattwert $2^{-230} \approx 5.8 \cdot 10^{-70}$ führten.

Bei vollständiger Berechnung sind 268.435.456 Blätter auszurechnen. Jeder Matrixeintrag ist

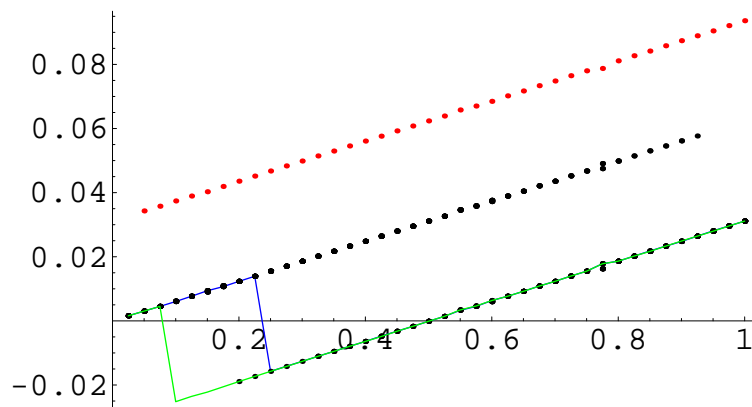


Bild 5.2: Die Entwicklung der Werte der ersten 16 Komponenten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

reell, deswegen sind auch alle Ergebnisvektorkomponenten reell. Abbildung 5.2 zeigt den Verlauf der 16 ersten der 1024 Komponenten. Die Werte sind abhängig von der aus den bereits errechneten Blättern aufsummierten Wahrscheinlichkeit. Dabei wurde die Wahrscheinlichkeit schrittweise um 0.025 erhöht, was zu 40 Messpunkten geführt hat. Der Verlauf zeigt drei gedachte steigende Linien. Auf der Mittleren Schwarzen bewegen sich zunächst einheitlich alle Komponenten. Interessant zu sehen ist das immer wieder auftretende „Umkippen“ einer einzelnen Komponente, beispielsweise sind die erste nach unten kippende Komponente (mit Index 13) grün und eine Weitere (mit Index 1) blau durchgezeichnet. Die nicht gesuchten Komponenten kippen nach unten auf die unterste Linie, die rote Linie, die den Verlauf der gesuchten fünften Komponente zeigt, ist bei 0.05 bereits nach oben gekippt. Die mittlere Linie endet, wenn jede Komponente umgekippt ist. Leider lässt sich nicht a-priori bestimmen, wann welche Komponente die Mittellinie verlässt. In diesem Beispiel etwa ist die gesuchte Komponente als Erste umgekippt. Begründet durch die in Abschnitt 4 getroffene Wahl der primalen Heuristik im Verfahren bestätigte sich jedoch bei mehrmaliger Ausführung der Simulation, dass voll und ganz der Zufall darüber entscheidet, wann welche Komponente die mittlere Linie verlässt.

Analog zu den einzelnen Komponenten kann die Entwicklung der Betragsquadrate der einzelnen Komponenten dargestellt werden. Wie in Bild 5.3 zu sehen ist, entspricht das hier der

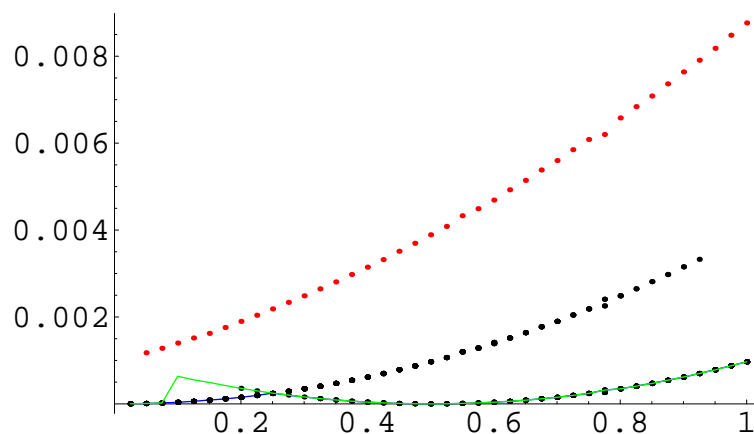


Bild 5.3: Die Entwicklung der Betragsquadrate der ersten 16 Komponenten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

Quadrierung der jeweiligen Werte und es sind wiederum die drei Linien zu erkennen. Die Idee dahinter war, zu einem Messzeitpunkt bei den Betragsquadraten möglicherweise Parallelen zu den Betragsquadraten des Endergebnisses (den tatsächlichen Wahrscheinlichkeiten) zu finden, z.B. im Quotient einzelner Komponenten zueinander. Dies bestätigte sich jedoch empirisch bei keiner der durchgeführten Simulationen.

Jedes Blatt liefert über die 28 Hadamard-Gates den gleichen Beitrag zur Gesamtwahrscheinlichkeit (Aufteilung in jedem Schritt: $\frac{1}{2} \wedge \frac{1}{2}$), nämlich $\frac{1}{2^{28}} \approx 3.7 \cdot 10^{-9}$. Dadurch ist der lineare Verlauf in Bild 5.4 zu erklären. In jedem Messschritt ist, um die berechnete Wahrscheinlichkeit um 0.025 erhöhen zu können, die Berechnung von etwa 6.710.900 Blättern nötig. Der Informationsverlust nach dem ersten Durchlauf des Branch&Bound-Verfahrens von der Wurzel zum Blatt liegt gemäß (3.10) bei nahezu 1, da nur $\frac{1}{2^{28}}$ der Gesamtwahrscheinlichkeit berechnet werden.

Fazit. Die vielfache gleichmäßige Aufteilung von Wahrscheinlichkeiten, wie bei der Hadamard-Transformation durchgeführt, ergibt Blattbeiträge mit einer sehr kleinen Wahrscheinlichkeit. Das Vergrößern bzw. Verkleinern von Amplituden der Ergebnisvektorkomponenten wird durch das Vorzeichen der Blattwerte bestimmt. Eine etwa gleiche Anzahl mit „+“ und „-“ ergibt einen kleinen Komponentenwert, ein großer Komponentenwert wird durch eine stärker voneinander differierende Zahl Blätter mit positivem bzw. negativem Vorzeichen gebildet. Damit ist es nicht möglich, mit wenigen Berechnungsschritten eine sehr gute Approximation des Gesamtergebnisses herbeizuführen. Das „Umkippen“ der Komponenten ist somit vergleichbar mit einem zufälligen Ereignis: Kippt eine Komponente nach oben, ist die Gesuchte gefunden.

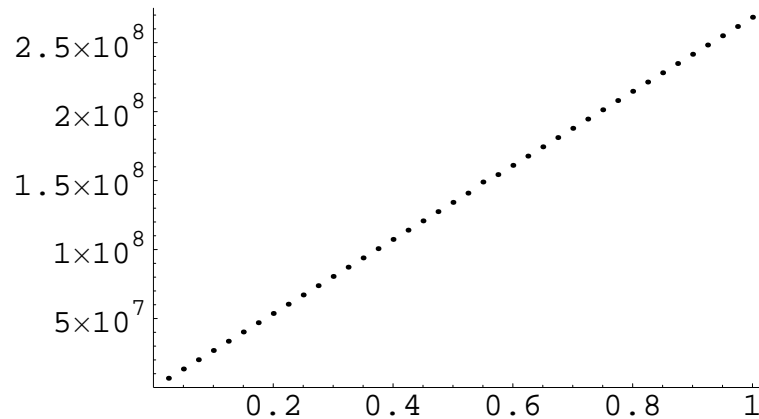


Bild 5.4: Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.

5.2 Quantum-Walk

Anhand eines Graphen G_2 soll ein Quantum-Walk gemäß 3.3.3 simuliert werden. Dabei sollen zwei verschiedene Kolorierungen wie in den Bildern 5.5 und 5.6 zu sehen angewendet werden. In der ersten Kolorierung werden die Farben grün, rot und orange jeweils sechsmal und violett

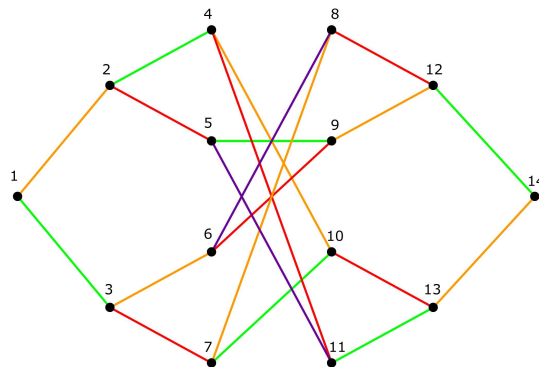
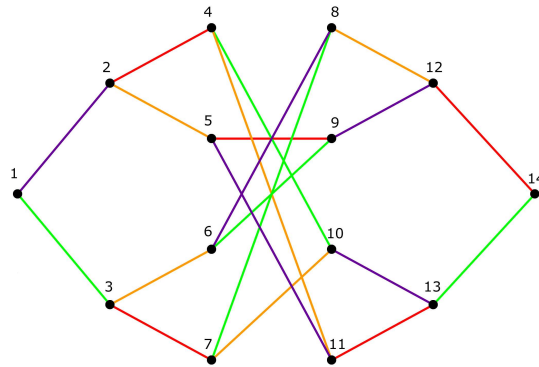
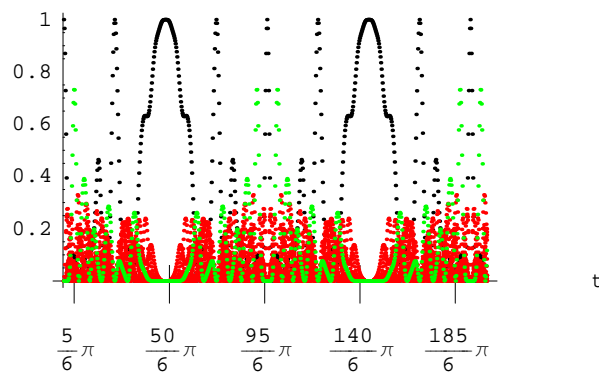


Bild 5.5: G_2 mit erster Kolorierung

zweimal verwendet, während in der Zweiten jede dieser vier Farben genau fünfmal benutzt wird. Der START-Knoten ist jeweils Knoten 1 und der gesuchte ENDE-Knoten ist Knoten 14. Für diese Problemstellung werden nach (3.21) für den QA 10 q-Bits benötigt. Die Diskretisierung des QA ist an Festlegungen der Variablen t und m geknüpft. Mit dem Parameter t lässt sich die Wahrscheinlichkeitsamplitude des gesuchten Knotens nach Durchlauf des QA einstellen. Es hat sich experimentell gezeigt, dass hier für eine beliebige Wahl von m der Wert $t = \frac{5}{6}\pi$ eine

Bild 5.6: G_2 mit zweiter Kolorierung

große Wahrscheinlichkeitsamplitude erzeugt (siehe beispielsweise in Bild 5.7: Grün ist der ENDE-, schwarz der START-Knoten, rot die Übrigen). In Bild 5.8 sind für die erste Kolorierung und Werte $t = \frac{5}{6}\pi$ bzw. $1 \leq m \leq 31$ die Wahrscheinlichkeitsamplituden der 14 Knoten zu sehen. Grün ist der ENDE-Knoten, schwarz der START-Knoten dargestellt. Erkennbar ist, dass sich für $m < 3$ die gesuchte Wahrscheinlichkeitsamplitude kaum von den anderen unterscheiden lässt. Um hier einen deutlichen Unterschied zu erhalten, wurde für eine erste Simulation $m = 4$ gewählt. Andererseits wird für große m gemäß [Chil02@] ein Sättigungspunkt erreicht (je größer t ist, desto später tritt der Sättigungspunkt ein). Deswegen soll eine zweite Simulation mit $m = 8$, wobei die Amplitude dem Sättigungswert nahe gekommen ist, durchgeführt werden. Die Farbmatrizen V_c sind Permutationsmatrizen von $\mathbb{1}_{1024}$. Zur Erzeugung eines solchen Gates kann etwa Mathematica wie in Bild 5.9 verwendet werden. Das Gate muss dann anschließend in Bit- m -Operationen zerlegt werden. Für jede Farbe wurden dafür jeweils Bit-9- bis Bit-2-Operationen, insgesamt also 8 Bit- m -Operationen je Farbe, benötigt. Damit sind die Black-Box-Matrizen erstellt. Für die bei fest gewählten t und m konstante Matrix T wird unabhängig von den Parametern die Diskretisierung mit 18 Bit- m -Operationen erreicht. Für $m = 4$ entstehen somit zusammen 544, für $m = 8$ insge-

Bild 5.7: Amplituden der Knoten für $m = 8$ und $0 \leq t \leq \frac{799}{24}\pi$.

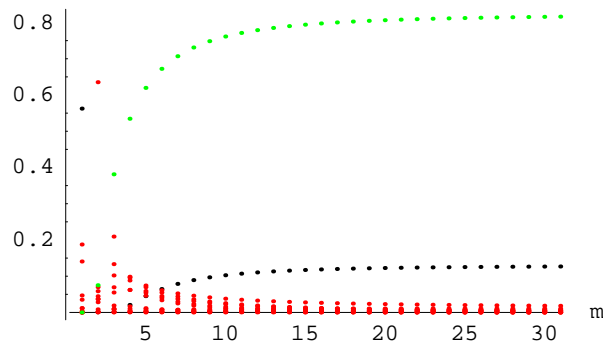


Bild 5.8: Amplituden der Knoten für $t = \frac{5}{6}\pi$ und $1 \leq m \leq 31$.

samt 1088 Bit- m -Operationen. Von Null verschiedene Werte der Ergebniskomponenten erhalten nur die den 14 Knoten entsprechenden Komponenten. Der Wert einer Komponente ist entweder stets reell oder stets imaginär. Deswegen lassen sich bereits diese Werte miteinander vergleichen.

```

Vrot = IdentityMatrix[1024] - IdentityMatrix[1024];
For[i = 0, i < 1024,
  If[BitAnd[i, 2^^0001010101] == 0, v[i] = 1 + BitXor[i, 2^^0110101010]];
  If[BitAnd[i, 2^^0001010101] == 1, v[i] = 1 + BitXor[i, 2^^0000001010]];
  If[BitAnd[i, 2^^0001010101] == 4, v[i] = 1 + BitXor[i, 2^^0000101000]];
  If[BitAnd[i, 2^^0001010101] == 5, v[i] = 1 + BitXor[i, 2^^0000000010]];
  If[BitAnd[i, 2^^0001010101] == 16, v[i] = 1 + BitXor[i, 2^^0010000000]];
  If[BitAnd[i, 2^^0001010101] == 17, v[i] = 1 + BitXor[i, 2^^0110101010]];
  If[BitAnd[i, 2^^0001010101] == 20, v[i] = 1 + BitXor[i, 2^^0000001000]];
  If[BitAnd[i, 2^^0001010101] == 21, v[i] = 1 + BitXor[i, 2^^0110101010]];
  If[BitAnd[i, 2^^0001010101] == 64, v[i] = 1 + BitXor[i, 2^^0000100000]];
  If[BitAnd[i, 2^^0001010101] == 65, v[i] = 1 + BitXor[i, 2^^0110101010]];
  If[BitAnd[i, 2^^0001010101] == 68, v[i] = 1 + BitXor[i, 2^^0010100000]];
  If[BitAnd[i, 2^^0001010101] == 69, v[i] = 1 + BitXor[i, 2^^0010100010]];
  If[BitAnd[i, 2^^0001010101] == 80, v[i] = 1 + BitXor[i, 2^^0010001000]];
  If[BitAnd[i, 2^^0001010101] == 81, v[i] = 1 + BitXor[i, 2^^0010001010]];
  If[BitAnd[i, 2^^0001010101] == 84, v[i] = 1 + BitXor[i, 2^^0110101010]];
  If[BitAnd[i, 2^^0001010101] == 85, v[i] = 1 + BitXor[i, 2^^0110101010]];
  Vrot[[v[i], i + 1]] = 1;
  i++];

```

Bild 5.9: Erzeugung der Farbgebungsmatrizen, hier V_{rot} der zweiten Kolorierung

Interpretation der Ergebnisse

Es gibt, wie in der rechten Spalte der Tabelle 5.2 zu sehen ist, für jedes Problem eine bestimmte Anzahl an Blättern zu berechnen. Es treten normalerweise keine Hadamard-Gates auf. Die Aufteilung der Wahrscheinlichkeiten erfolgt wie in Kapitel 4 dargestellt in Form einer binären Verzweigung entweder mit $1 \wedge 0$ oder $0.62941 \wedge 0.37059$ für $m = 4$ bzw. entweder mit $1 \wedge 0$ oder $0.89668 \wedge 0.10332$ für $m = 8$. Die Blattwerte liefern ausschließlich Beiträge zu den zu allen Knoten korrespondierenden Komponenten des Ergebnisvektors. Für die hier durchgeführten Quantum-Walks sind das stets die Basisvektoren $e_1, e_2, e_5, e_6, e_{17}, e_{18}, e_{21}, e_{22}, e_{65}, e_{66}, e_{69}, e_{70}, e_{81}$ und e_{82} .

Der größte Blattbeitrag zu einer Komponente beträgt 0.02463 ($m = 4$) bzw. 0.1747 ($m = 8$). Die Bilder 5.10, 5.14, 5.18 und 5.22 zeigen den jeweiligen Verlauf der Ergebniskomponenten in Abhängigkeit der erklärten Gesamtwahrscheinlichkeit. Dabei sind diejenigen Komponenten, deren Wert rein imaginär ist, blau dargestellt, die rein reellen Komponenten dagegen schwarz. Die gesuchte Komponente - der Knoten 14 - ist in rot gezeichnet. Diese Komponente ist ebenfalls reell. Die Werte wurden - wie auch bei den folgenden Plots der Betragsquadrate - zur besseren Veranschaulichung als durchgezogene Linien dargestellt. Interessant zu sehen ist der unterschiedliche Verlauf der Komponentenwerte. Die rein reellen Werte entwickeln sich in ähnlicher Weise, aber sich deutlich unterscheidend zu den rein imaginären Werten, die sich innerhalb ihrer Gruppe ebenfalls vergleichbar entwickeln. Lediglich die rote Linie passt sich nicht ganz in das Gesamtgefüge ein.

Die Abbildungen 5.11, 5.15, 5.19 und 5.23 veranschaulichen die Entwicklung der Betragsquadrate der 14 Komponenten, wobei die Farbgebung gleich den Komponenten-Abbildungen gewählt wurde. Es lässt sich keine signifikante Besonderheit bei der roten Linie sehen. Während sie in den ersten beiden dieser Plots ($m = 4$) sich noch deutlich von den Anderen abhebt, sind in den letzten beiden Plots ($m = 8$) keine Besonderheiten zu erkennen. Die Probleme mit $m = 8$ unterscheiden sich jedoch noch weiter: Hierbei sind sogar in der Darstellung der Betragsquadrate in den Bildern 5.19 und 5.23 die imaginären und reellen Komponenten deutlich unterscheidbar. Die rote Linie lässt sich dennoch nicht zuordnen.

In den Streudiagrammen 5.12, 5.16, 5.20 und 5.24 ist jeweils die an den Messpunkten erreich-

m, t	Kolorierung (Abb.)	Abbildungen (Ko, Be, Bl, Su)	Anz. Blätter
$4, \frac{5}{6}\pi$	5.5	5.10, 5.11, 5.12, 5.13	3.392
$4, \frac{5}{6}\pi$	5.6	5.14, 5.15, 5.16, 5.17	4.520
$8, \frac{5}{6}\pi$	5.5	5.18, 5.19, 5.20, 5.21	11.174.288
$8, \frac{5}{6}\pi$	5.6	5.22, 5.23, 5.24, 5.25	19.001.866

Tafel 5.2: Die vier Simulationen des Quantum-Walk mit den Abbildungen zu den Komponenten (Ko), Betragsquadraten der Komponenten (Be), berechneten Blättern (Bl) und aufsummierten Wahrscheinlichkeiten (Su) und den zu berechnenden Blättern.

te Wahrscheinlichkeit auf der x-Achse gegen die Anzahl der dazu zu berechnenden Blätter auf der y-Achse geplottet. Das sich steigernde Wachstum der benötigten Blätter zum Erreichen der nächsten Wahrscheinlichkeitsstufe ist deutlich zu sehen. Insbesondere die Plots für $m = 8$ zeigen, dass zum Erreichen von etwa 0.9 der Gesamtwahrscheinlichkeit im Vergleich zur Gesamtzahl an Blättern äußerst wenige berechnet werden müssen (etwa 1‰). Ebenfalls in Streudiagrammen (5.13, 5.17, 5.21 und 5.25) ist wiederum jeweils die an den Messpunkten erreichte Wahrscheinlichkeit auf der x-Achse gegen nun die aufsummierten Betragsquadrate der 14 Komponenten

geplottet. Hier ist vor allen Dingen interessant, dass sich die Betragsquadrate nicht zu jedem Zeitpunkt zu der erreichten Wahrscheinlichkeit aufsummieren, in diesem Plot somit kein linearer Verlauf erkennbar ist. Das Korollar 4.2 wird damit durch die Simulation bestätigt. Tafel 5.2 zeigt zusammengefasst die zu den einzelnen Quantum-Walk-Problemen gehörenden Abbildungen.

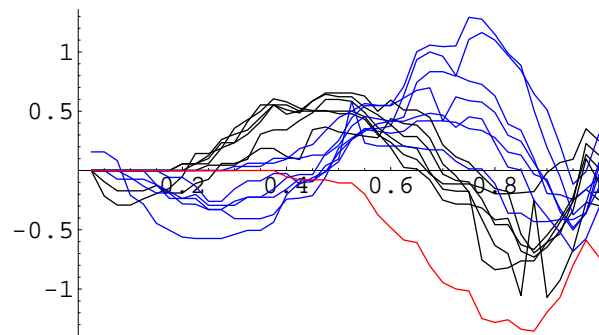


Bild 5.10: Entwicklung der Werte der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

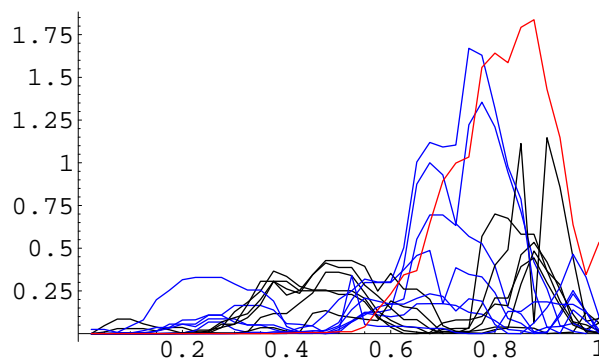


Bild 5.11: Entwicklung der Betragsquadrate der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

Der Informationsverlust nach einem Durchlauf des Baumes von der Wurzel zum Blatt liegt gemäß (3.10) im Mittel etwa bei 0.9995 ($m = 4$) bzw. 0.99 ($m = 8$).

Fazit. Die sehr stark unterschiedliche Aufteilung der Wahrscheinlichkeiten führt zu gewünschten Effekten. Wenige Blätter liefern einen sehr großen Beitrag zur Gesamtwahrscheinlichkeit. Bereits nach einem Durchlauf zeigt sich mit steigender Iterationszahl m ein geringer werdender Informationsverlust. Aber es kommt im Verlauf des Verfahrens zu gewaltigen Verzerrungen des Ergebnisvektors. Betragsquadrate über dem Wert 10 (vgl. Bild 5.19) sind nicht zu vermeiden. Das lässt aber keinen Spielraum in Bezug auf eine Wahrscheinlichkeitsinterpretation.

Auffällig ist jedoch die sich von den anderen Komponenten abweichend verhaltende gesuchte Komponente. Sie lässt sich grafisch - trotz rein reellen Wertes - weder den reellen Komponenten

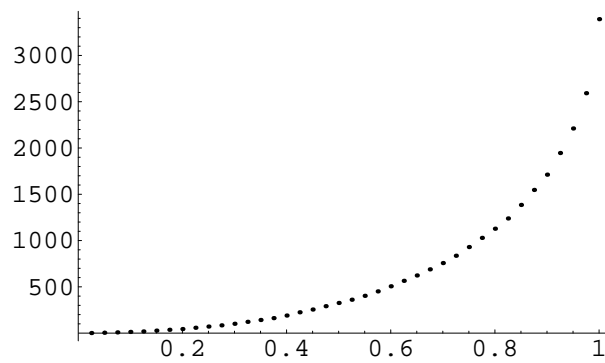


Bild 5.12: Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.

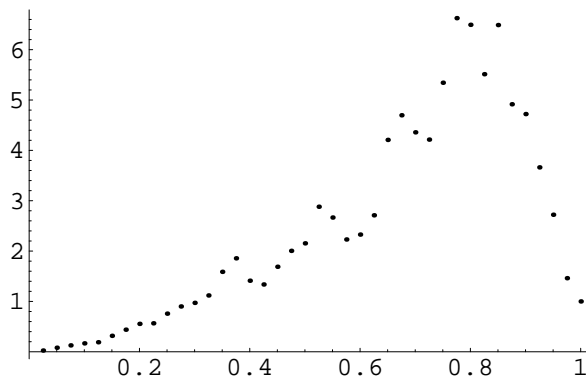


Bild 5.13: Scatterplot: $y \leftrightarrow x$: Aufsummierte Betragsquadrate der 14 Komponenten \leftrightarrow berechnete Wahrscheinlichkeit.

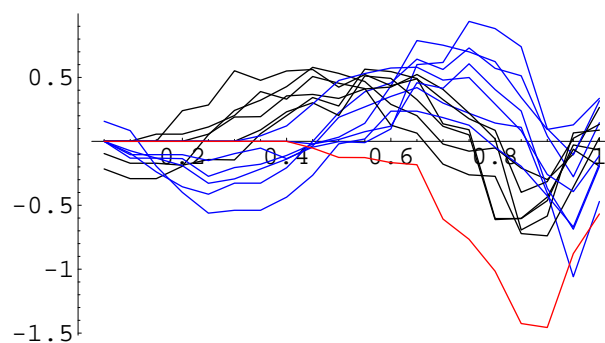


Bild 5.14: Entwicklung der Werte der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

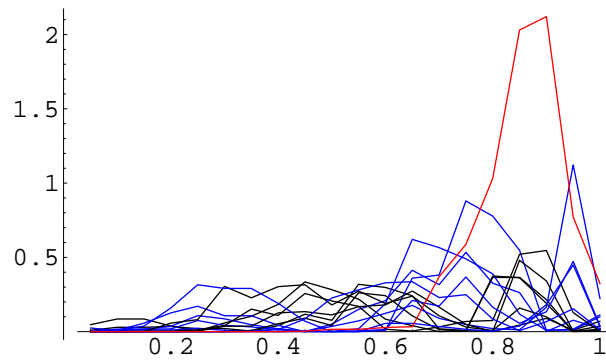


Bild 5.15: Entwicklung der Betragsquadrate der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

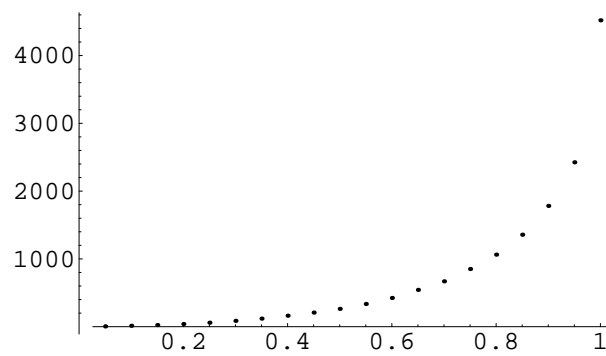


Bild 5.16: Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.

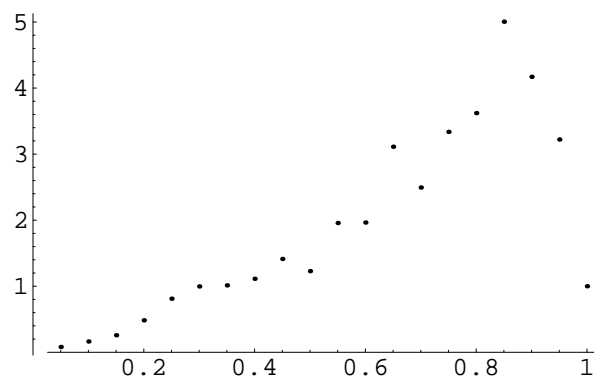


Bild 5.17: Scatterplot: $y \leftrightarrow x$: Aufsummierte Betragsquadrate der 14 Komponenten \leftrightarrow berechnete Wahrscheinlichkeit.

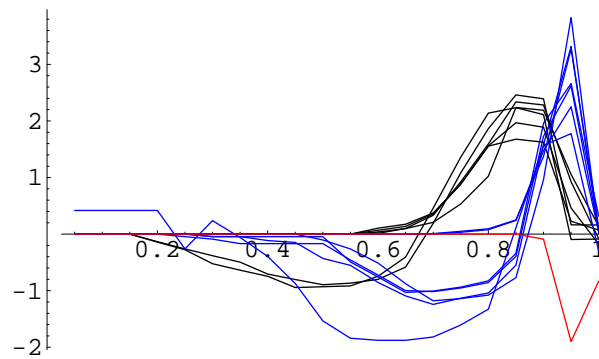


Bild 5.18: Entwicklung der Werte der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

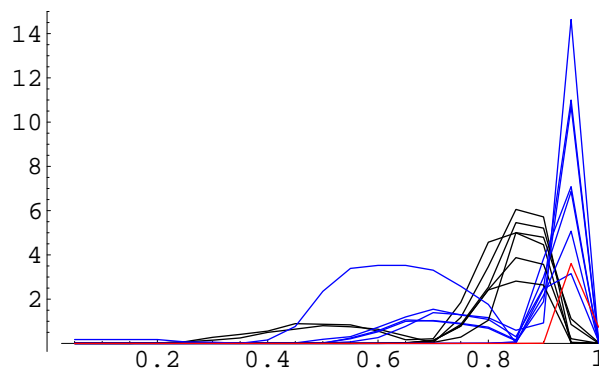


Bild 5.19: Entwicklung der Betragsquadrate der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

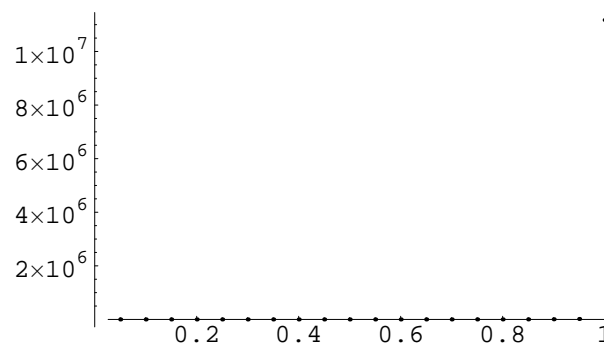


Bild 5.20: Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.

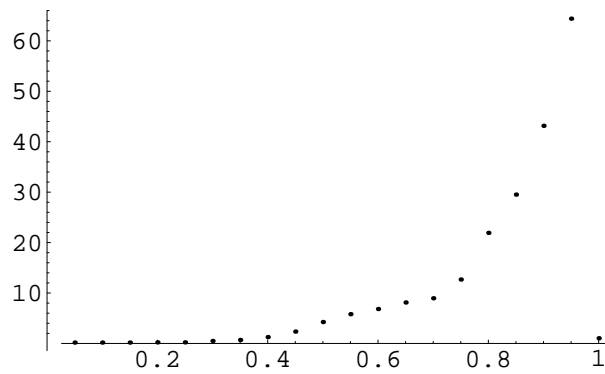


Bild 5.21: Scatterplot: $y \leftrightarrow x$: Aufsummierte Betragsquadrate der 14 Komponenten \leftrightarrow berechnete Wahrscheinlichkeit.

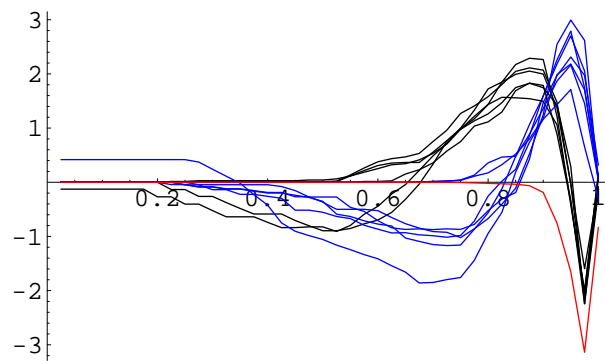


Bild 5.22: Entwicklung der Werte der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

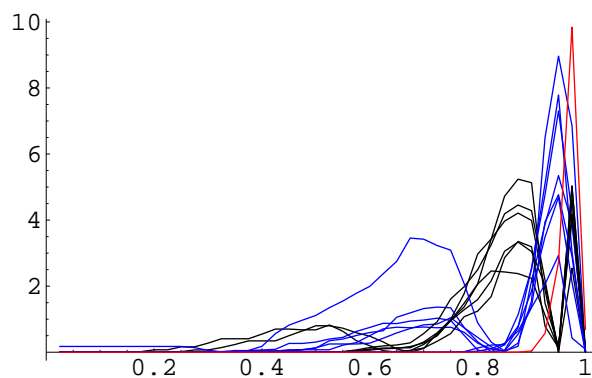


Bild 5.23: Entwicklung der Betragsquadrate der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.

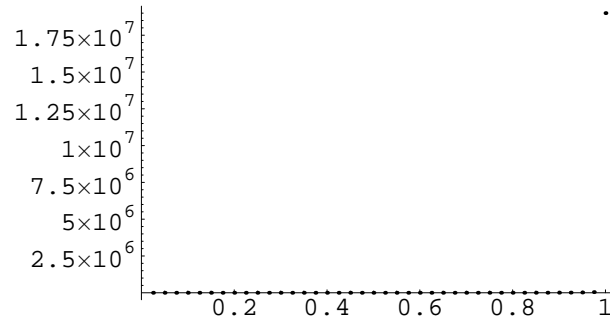


Bild 5.24: Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.

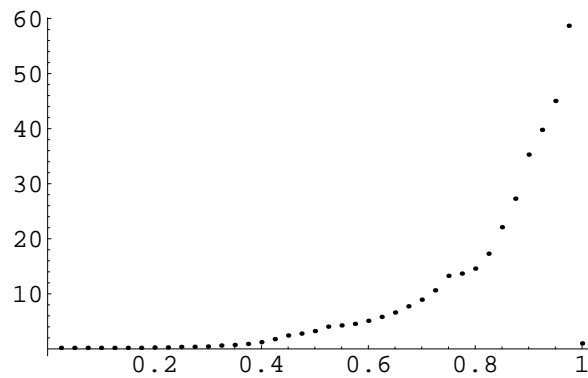


Bild 5.25: Scatterplot: $y \leftrightarrow x$: Aufsummierte Betragsquadrate der 14 Komponenten \leftrightarrow berechnete Wahrscheinlichkeit.

noch den rein imaginären Komponenten zuordnen. Hier ist ein Ansatzpunkt zum Auffinden der gesuchten Komponente. Dies könnte in hierauf aufbauenden Studien untersucht werden.

KAPITEL 6

Zusammenfassung und Ausblick

Quantenalgorithmen als stochastische Algorithmen

Aufbauend auf den bekannten Grundlagen des auf quantentheoretischen Zusammenhängen basierenden QC konnte eine Studie zur Simulation von QA durchgeführt werden. Da dabei QA auf vorhandener Technologie lauffähig gemacht und darüber hinaus auf eine Vergleichbarkeit des Berechnungsaufwandes geachtet werden muss, ist einiges an Vorarbeit notwendig.

Zunächst wurden Bit-m-Operationen definiert, die dafür sorgen, dass jede quantentheoretische Auswertung den gleichen klassischen Aufwand erzeugt. Es wurde erfolgreich gezeigt, dass mit diesen Operationen jedes beliebige Gate mit Bit-m-Operationen dargestellt werden kann. Dazu wurde auch eine systematische algorithmische Zerlegung eines gegebenen Gates entwickelt, die eine Obergrenze für die dafür benötigten Bit-m-Operationen liefert.

Die immense Komplexität von QA muss reduziert werden. Eine Möglichkeit hierfür ist, je Auswertungsschritt den Rechenaufwand zu minimieren. Jede Anwendung eines Gates bestimmt eine neue Verteilung der Wahrscheinlichkeiten der Basisvektoren. Diese stochastische Struktur eines QA lässt die Vermutung aufkommen, es mit einem Algorithmus, der mit stochastischen Algorithmen vergleichbar ist, zu tun zu haben. Das wird aber nur erreicht, wenn nicht in jedem Schritt eine vollständige Auswertung vollzogen wird, sondern lediglich ein Basisvektor in einer Einzelauswertung mit einer Gate-Komponente neu gewichtet wird. Das reduziert zwar die Komplexität auf ein Minimum, erzeugt andererseits aber einen starken Informationsverlust, der durch zusätzliche Einzelauswertungen unter einer Komplexitätssteigerung sukzessive zurückgeführt werden kann. Damit ist auch eine Vergleichbarkeit zwischen einer Auswertung in der Quantenwelt und der klassischen Welt erreicht.

Klassifikation von Quantenalgorithmen

Die vorhandenen QA besitzen unterschiedliche Techniken, derer sie sich bedienen. Dabei haben sich bis heute drei wesentliche Techniken herauskristallisiert. Auf Grund der drei Techniken kann eine Klassifikation von QA erstellt werden. Das führt zu den Klassen der Hidden-Subgroup-, amplitudenverstärkenden und der Quantum-(Random)-Walk-Probleme. Wegen dieser Klassifikation genügt jeweils die Untersuchung eines Vertreters einer Klasse, um herauszufinden, ob sich eine Simulation im Hinblick auf eine Reduktion der Komplexität für die Vertreter der jeweiligen Klasse lohnt.

Um die Simulation durchführen zu können, ist zunächst die Modellierung des stochastischen Algorithmus notwendig. Dies kann in Form eines Branch&Bound-Verfahrens auf einem Binärbaum erreicht werden. Es wurde dazu ein JAVA-Tool implementiert, das einen QA, dessen Gates aus

Bit-m-Operationen besteht, mittels der beschriebenen stochastischen Einzelauswertungen durchgeführt und statistische Ergebnisse zur Aufwandsbestimmung liefert.

Die mit dem JAVA-Tool erzielten Simulationsergebnisse zeigen, dass die QA der Amplitudenverstärkung und damit auch die Hidden-Subgroup-QA, die mit einer ähnlichen Technik arbeiten, keine Reduktion der Komplexität liefern. Die Quantum-(Random)-Walk-QA jedoch liefern mögliche Ansatzpunkte zur Reduktion der Komplexität. Denn mit wenig Rechenaufwand kann bereits sehr viel an Information bestimmt werden.

Perspektiven bei der klassischen Umsetzung

Die wichtige Grundlage der Vergleichbarkeit von QA mit klassischen Algorithmen ist gelegt. Um deren Möglichkeiten, ohne dafür Quantenrechner bauen zu müssen, nutzen zu können, muss es möglich gemacht werden, Komplexitätsgewinne auf klassischen Rechnern nicht wieder zu verlieren. Die mit einem programmierten Tool zur Simulation von Quantenalgorithmen auf vorhandener Rechnertechnologie durchgeführten Berechnungen zeigen eine Möglichkeit dazu. Bei den frühen Quantenalgorithmen gibt es damit wohl keine Möglichkeit, Effizienzvorteile erreichen zu können. Bei jüngeren und mit neuen Techniken aufgestellten Algorithmen jedoch werden Ansatzpunkte dargestellt, mit denen es möglich sein könnte, den gegebenen Vorteil zu erhalten. Die Chancen ruhen auf neuen Techniken, wobei dabei Fall für Fall getestet werden muss. Die Erfahrungen dieser Arbeit zeigen, dass abhängig von den verwendeten Techniken jeweils ein anderes Verhalten bei der Simulation zu beobachten ist. Es ist nun aber möglich, gegebene Quantenalgorithmen in die für das Tool vorgegebene Form zu bringen und dann nach Wunsch Simulationen nach eigenen Kriterien durchzuführen.

Abbildungsverzeichnis

1	Komplexität vs. Informationsverlust	8
3.1	QFT in Form von Bit-m-Operationen.	52
3.2	Beispiel eines Graphen G_4	54
4.1	Interpretation des Verzweigungsprozesses als Binärbaum	58
5.1	Screenshot des Branch&Bound JAVA-Programms	71
5.2	Die Entwicklung der Werte der ersten 16 Komponenten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	73
5.3	Die Entwicklung der Betragsquadrate der ersten 16 Komponenten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	74
5.4	Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.	75
5.5	G_2 mit erster Kolorierung	75
5.6	G_2 mit zweiter Kolorierung	76
5.7	Amplituden der Knoten für $m = 8$ und $0 \leq t \leq \frac{799}{24}\pi$	76
5.8	Amplituden der Knoten für $t = \frac{5}{6}\pi$ und $1 \leq m \leq 31$	77
5.9	Erzeugung der Farbgebungsmatrizen, hier V_{rot} der zweiten Kolorierung	77
5.10	Entwicklung der Werte der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	79
5.11	Entwicklung der Betragsquadrate der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	79
5.12	Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.	80
5.13	Scatterplot: $y \leftrightarrow x$: Aufsummierte Betragsquadrate der 14 Komponenten \leftrightarrow berechnete Wahrscheinlichkeit.	80
5.14	Entwicklung der Werte der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	80
5.15	Entwicklung der Betragsquadrate der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	81
5.16	Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.	81
5.17	Scatterplot: $y \leftrightarrow x$: Aufsummierte Betragsquadrate der 14 Komponenten \leftrightarrow berechnete Wahrscheinlichkeit.	81

5.18	Entwicklung der Werte der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	82
5.19	Entwicklung der Betragsquadrate der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	82
5.20	Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.	82
5.21	Scatterplot: $y \leftrightarrow x$: Aufsummierte Betragsquadrate der 14 Komponenten \leftrightarrow berechnete Wahrscheinlichkeit.	83
5.22	Entwicklung der Werte der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	83
5.23	Entwicklung der Betragsquadrate der 14 Knoten des Ergebnisvektors in Abhängigkeit der in insgesamt 40 Messpunkten bestimmten Gesamtwahrscheinlichkeit.	83
5.24	Scatterplot: $y \leftrightarrow x$: Anzahl berechneter Blätter \leftrightarrow berechnete Wahrscheinlichkeit.	84
5.25	Scatterplot: $y \leftrightarrow x$: Aufsummierte Betragsquadrate der 14 Komponenten \leftrightarrow berechnete Wahrscheinlichkeit.	84

Tabellenverzeichnis

1	Vektoren- und Dirac-Notation im Vergleich [Pere93].	1
2.1	Partitionierung der Zustände nach dem m -ten Bit	26
2.2	Die ersten 7 Transformationen zur Zerlegung der Beispielmatrix.	36
2.3	Bit- m -Operationen und deren Operationsbits der Matrix aus Beispiel 2.3.3 der Reihenfolge nach geordnet, $M = R_1 \cdot R_2 \cdot R_3 \cdot R_4 \cdot R_5$	37
3.1	Ausgewählte Symbole eines Quantenschaltkreises	44
5.1	Dateigrößen ausgewählter Datendateien der einzelnen Probleme	72
5.2	Die vier Simulationen des Quantum-Walk mit den Abbildungen zu den Komponenten (Ko), Betragsquadraten der Komponenten (Be), berechneten Blättern (Bl) und aufsummierten Wahrscheinlichkeiten (Su) und den zu berechnenden Blättern.	78

Literaturverzeichnis

- [Ahar03] Adiabatic Quantum State Generation and Statistical Zero Knowledge,
D. Aharonov, STOC03, San Diego, California, USA, 2003
- [Aulb96] Analysis III,
B. Aulbach, 1996 (Vorlesungsskript)
- [Aume03@] Der SHOR-Algorithmus,
A. Aumer, 2003
- [Baue02] Quantengatter & -Algorithmen,
S. Bauer, 2002
- [Borg01] Optimierung, Operation Research und Spieltheorie,
K.H. Borgwardt, Birkhäuser-Verlag 2001
- [Born03] Quantenrechnen,
F. Bornemann, 2003 (Vorlesungsskript)
- [Bouw00] The Physics of Quantum Information,
Bouwmeester, Ekert, Zeilinger, 2000
- [Bras00@] Quantum Amplitude Amplification and Estimation,
G. Brassard, arXiv:quant-ph/0005055, 2000
- [Chil00] Quantum Information Processing in Continuous Time,
A.M. Childs, California Institute of Technology, 2000
- [Chil02@] Exponential algorithmic speedup by quantum walk,
A.M. Childs, Massachusetts Institute of Technology, arXiv:quant-ph/0209131,
2002
- [Däne02] Die verrückte Welt der Quantencomputer,
B. Däne, TU Ilmenau, 2002
- [Deuf93] Numerische Mathematik I,
P. Deuffhard, A. Hohmann, de Gruyter 1993
- [Feyn85] Quantum mechanical computers,
R.P. Feynmann, Optics News 11, 11-20, 1985
- [Hebe00] Simulation of Quantum Mechanical Problems on Classical and Quantum Mechanical Computers,
C. Hebeisen, 2000

- [Heis79] Quantentheorie und Philosophie,
Reclam Stuttgart 1979
- [Josz03@] Notes on Hallgrens efficient quantum algorithm for solving Pells equation,
R. Josza, arXiv:quant-ph/0302134 v1, 2003
- [Kemp04] Quantum Walks: An approach to quantum computing,
J. Kempe, UC Berkeley and CNRS & LRI, Univ. de Paris-Sud, Orsay, France,
2004.
- [Kieu01@] Quantum Algorithm for Hilberts Tenth Problem,
Tien D Kieu, arXiv:quant-ph/0110136, 2001
- [Kita95@] Quantum measurements and the abelian stabilizer problem,
A. Kitaev, arXiv:quant-ph/9511026, 1995
- [Lato03@] Adiabatic quantum computation and quantum phase transitions,
J.I. Latorre, arXiv:quant-ph/0308042, 2003
- [Lomo02@] Quantum Hidden Subgroup Algorithms: A Mathematical Perspective,
S.J. Lomonaco, arXiv:quant-ph/0201095 v3, 2002
- [Lomo04@] Quantum Hidden Subgroup Algorithms: The Devil Is in the Details,
S.J. Lomonaco, arXiv:quant-ph/0403229 v1, 2004
- [Math04] Quantum Computation und stochastische Algorithmen (Teil 1),
Mathematical Engineering GmbH, 2004
- [MeiS05] Stochastik, Theorie und Anwendungen,
D. Meintrup, S. Schäffler, Springer 2005
- [Meye73] Meyers Physik Lexikon,
Bibliographisches Institut Mannheim 1973
- [Mosc99@] The Hidden Subgroup Problem and Eigenvalue Estimation on a Quantum Com-
puter,
M.Mosca, arXiv:quant-ph/9903071 v1, 1999
- [Münn02@] Quantentheorie,
G. Münster, [http://www.pauli.uni-muenster.de/
menu/Lehre/quant-skript/skriptum-h.html](http://www.pauli.uni-muenster.de/menu/Lehre/quant-skript/skriptum-h.html), 2002
- [Müth99] Quantenmechanik I und II,
H. Müther, 1999 (Vorlesungsskript)
- [Niel00] Quantum Computation and Quantum Information, M.A. Nielsen, I.L. Chuang,
Cambridge Press 2000
- [Pere93] Quantum Theory: Concepts and Methods,
A. Peres, Kluwer Academic Publishers, 1993
- [Qcir@] <http://info.phys.unm.edu/Qcircuit>
- [Råde97] Mathematische Formeln,
L. Råde, B. Westergren, Springer 1997

-
- [Schä02] Quantum Computation,
S. Schäffler, 2002
- [Suto03] Ein stochastisches Verfahren zur globalen Optimierung bei diskreten und kontinuierlichen Variablen,
A. Sutor, 2003

Index

A

Algorithmus 9, 57, 60
 Stochastischer 8, 46, 56–58
 Zerlegungs- 38

B

Binärbaum 8, 57
Bit 6, 11, 20
 -kombination 25, 29
 Multi-q- 7, 22, 40
 q- 21, 22, 40
Black-Box 37, 42, 53, 54
Blatt 54, 57, 60, 69
Branch&Bound-Verfahren 57, 59, 69

D

Dirac-Notation 1

E

Entanglement 25

G

Gate 7, 26, 40, 43
 Controlled- 51
 Diagonalisierung 33
 Hadamard- 45
Graph 54, 54, 56

H

Hilbertraum 4, 13, 18, 40

I

Informationsverlust 7, 48, 57, 74, 79

J

JAVA 9, 71

K

Kante 53, 54
 Wahrscheinlichkeit 58
Knoten 8, 53, 58
 Wurzel- 57
Komplexität 7, 48, 57
Kopenhagener Deutung 18
Korrespondenz 28, 34

M

Messung 8, 15, 22, 24
Modell 43
 (zeit)diskretes 8, 43, 45
 (zeit)kontinuierliches 8, 43, 45

N

Notationen 1

O

Observable 15
 verträglich 15
Operation 26
 Bit-m- 28, 35, 40, 42, 71
 Ein-Bit- 8, 40
Operator 14, 43
 Adjungierter 14
 Eigenwert 14, 15
 Eigenzustand 14, 15
 Hamilton- 16, 18, 42, 43, 48
 Hermitescher 14
 Linearer 14
 Selbstadjungierter 15, 18
 Zeitentwicklungs- 16, 18, 25

P

Partitionierung 24
 Bit-m- 27, 28

Pfad	57, 60
Postulat	10, 20, 25, 43
1-tes	12
2-tes	15
3-tes	18
Prozess	
Stochastischer	46
Verzweigungs-	58

Q

Quantenalgorithmus	6, 42, 43, 56
Adiabatischer	42, 45, 45
Amplitudenverstärkung	8, 42, 52, 53, 72
Charakteristika	8, 42
Diskretisierung	43, 48
Grover-	45, 53, 72
Hidden-Subgroup	8, 42, 50, 51
Quantum-(Random)-Walk	8, 42, 53, 54, 72, 75
Schaltkreis	42–45, 57
Shor-	6, 51, 70
Quantenmechanik	7, 10, 18, 20
Quantum Computation	6, 18, 20, 21, 43
Quantum Fourier Transformation	42, 51, 51

S

Schrödinger-Gleichung	10, 16, 43
zeitabhängige	16, 18
Simulation	7, 57, 71
Spin	6, 11
-down	10, 11, 12
-up	10, 11, 12
Teilchen-	10, 11, 13, 18
Superposition	13, 22, 69
-sprinzip	11, 18, 21
System	6
konservatives	16, 25, 40
n-Bit-	22, 26, 40

T

Transformationsmatrix	7, 33, 34
-----------------------------	-----------

U

unitär	7, 16, 18, 25, 43
Matrix	<i>siehe</i> Gate

V

Vektor	13, 20
Basis-	21, 23, 40
bra-	13
ket-	13
Verschränkung	<i>siehe</i> Entanglement

W

Wahrscheinlichkeitsraum	5
Welle-Teilchen-Dualismus	6, 10
Wellenfunktion	10, 11
Spinor-	11, 18, 21

Z

Zähldichte	24
Zufallsexperiment	4
Grundbegriffe	4
Zufallsvariable	5
Zustand	11, 18, 20
Klassischer	21
Reiner	13, 23
Spin-	12
Start-	42, 44
Stationärer	10
Zustandsraum	21, 22
Mathematischer	20
Physikalischer	10, 12
Stochastischer	46
Zustandsreduktion	15, 24