Hauptseminar Quantencomputer: Suche in Datenbanken

Thomas Reinberger

18.05.2003

Zusammenfassung

1994 hat Shor [1] gezeigt, dass Quantencomputer klassische Probleme wie die Faktorisierung von ganzen Zahlen im Gegensatz zu probabilistischen Maschinen effizient lösen können. In den folgenden Seiten wird gezeigt, dass es auch für die Problemklasse der Suchprobleme Algorithmen für Quantencomputer gibt, die wesentlich schneller arbeiten als Algorithmen auf probabilistischen Maschinen könnten. Es wird gezeigt, dass die Komplexität für dieses Suchproblem bei $O(\sqrt{N})$ liegt.

Inhaltsverzeichnis

1 Grundlegende Operationen

Suchprobleme lassen sich im Wesentlichen wie folgt beschreiben: Ein System habe $N = 2^n$ Zustände $S_1, S_2, ..., S_N$. Diese 2^n Zustände werden als n Bit-Zeichenketten repräsentiert. Es gebe einen eindeutigen Zustand S_v , so daß die Bedingung $C(S_v) = 1$ und $\forall\, S_i,\, i \neq v\, : C(S_i) = 0$. Der Zustand S_v soll identifiziert werden, weiterhin sei C ein Orakel. Grover's Algorithmus geht von einer unstrukturierten Anordnung der Zustände S aus. Der effizienteste klassische Algorithmus versucht, jeden Zustand zu testen. Um mit einer Wahrscheinlichkeit von mindestens $\frac{1}{2}$ den gesuchten Zustand S_v zu finden, sind trivialerweise mindestens $\frac{N}{2}$ Schritte notwendig. Generell nimmt man an, dass das Problem in $O(N)$ Schritten mit Sicherheit gelöst wird, da es N Schritte gibt. Da quantenmechanische Systeme aber in mehreren Zustände gleichzeitig sind, können mehrere Aufgaben gleichzeitig erledigt werden. Im folgenden wird auf die von Grover's Algorithmus benötigten Operationen eingegangen.

1.1 Initialisierung der Systemkonfiguration

Grover benutzt ein System, in dem die Amplituden aller s^N Zustände gleich sind. Zunächst wird ein Münzwurf auf ein einzelnes Bit ausgeführt, das die Zustände 0 und 1 haben kann. Die Münzwurf-Operation wird durch die Matrix $M = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ dargestellt (Hadamard-Transformation). Sei b der Zustand des Bits. Dadurch ergeben sich die Superpositionen

$$Mb = \begin{cases} \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) & falls\; b = 0 \\ \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right) & falls\; b = 1 \end{cases}$$

Der Unterschied zwischen den Superpositionen liegt in der Phase der Amplitude, die sich um das Vorzeichen unterscheidet. Die Operation M kann auf alle n Bits unabhängig voneinander angewandt werden. Wenn der Anfangszustand ψ_0 alle n Bits im Zustand 0 hatte, wird der Endzustand in allen 2^n Zuständen die gleiche Amplitude $2^{-\frac{n}{2}}$ haben. Wenn der Anfangszustand ψ_0 einen der 2^n Zustände hat, der aus einer n-Bit Zeichenkette dargestellt wird, die aus 0'en und 1'en besteht ergibt sich nach der Transformation mit M auf jedes Bit der Zeichenkette eine Superposition aus allen 2^n Zuständen, von denen jeder die Amplitude $-2^{-\frac{n}{2}}$ oder $2^{-\frac{n}{2}}$ hat.

1.2 Die Walsh-Hadamard-Transformation

Die Walsh-Hadamard-Transformation W entspricht der Hadamard-Transformation auf jedes einzelne QuBit:

$$W_2\,|0\rangle = \tfrac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle)$$
$$W_2\,|1\rangle = \tfrac{1}{\sqrt{2}}\,(|0\rangle - |1\rangle)$$

Die Anwendung von W auf ψ_0 liefert:

$$\overbrace{M \otimes ... \otimes M}^{W} \quad |\psi_0\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle_1 + |1\rangle_1) \otimes ... \otimes \tfrac{1}{\sqrt{2}}(|0\rangle_n + |1\rangle_n)$$

Da das Tensorprodukt linear ist, ergibt sich

$$W_N |\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0\cdots 0}^{1\cdots 1} |x\rangle$$

. Daraus ergibt sich die Amplitude von $\frac{1}{\sqrt{n}}$ für alle Zustände.

1.3 Selektive Rotation

Die selektive Rotation der Phase einer Amplitude für ein 4-Zustands-System hat die Form

$$\begin{bmatrix} e^{j\phi_1} & 0 & 0 & 0 \\ 0 & e^{j\phi_2} & 0 & 0 \\ 0 & 0 & e^{j\phi_3} & 0 \\ 0 & 0 & 0 & e^{j\phi_4} \end{bmatrix}$$

wobei $i = \sqrt{-1}$ und $\phi_1, \phi_2, \phi_3, \phi_4$ beliebige reele Zahlen sind. Bei der selektiven Rotation bleiben die Wahrscheinlichkeiten in den Zuständen erhalten, da das Quadrat des Betrags der Amplitude in jedem Zustand gleich bleibt.

Alternativ zur selektiven Rotation kann man in Grover's Algorithmus auch die Phaseninversion anwenden, die sogar noch weniger Rechenschritte erfordert.

2 Grover's Algorithmus

> Die Grundidee des Algorithmus von Grover ist es, die Amplitude des gesuchten Zustandes so weit zu erhöhen, bis die Wahrscheinlichkeit, den gesuchten Zustand zu messen, ausreichend groß ist.

Diese Idee lässt sich jedoch nur umsetzen, wenn eine Superposition von Zuständen vorliegt. Wenn sich das System in einem Zustand $|s\rangle$ befindet, führt folgende Idee zum Ziel:

> Wende die Selektive Rotation solange auf $|s\rangle$ an, bis wir beim Endzustand $|t\rangle$ mit $C(S) = 1$ sind.

1. Initialisiere das System auf die Verteilung $\left(\frac{1}{\sqrt{N}}, ..., \frac{1}{\sqrt{N}}\right)$. Diese Verteilung kann in $O(\log N)$ Schritten erreicht werden.
2. Wiederhole die folgende unitäre Operation $O(\sqrt{N})$ mal:
   - Das System sei in einem beliebigen Zustand S: Falls $C(S) = 1$ rotiere die Phase um π.

- Wende die Diffusions-Transformierung D auf den Zustand an. Die Diffusions-Transformierung definiert sich durch folgende Matrix: $D_{ij} = \frac{2}{N}\ falls\ i \neq j\ und\ D_{ii} = -1 + \frac{2}{N}$. Diese Matrix kann man durch $D = WRW$ implementieren, wobei W die Walsh-Hadamard-Transformation und R die Rotationsmatrix wie folgt definiert sind:

$$\begin{matrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & -1 \end{matrix}$$

  $W_{ij} = 2^{-\frac{n}{2}}(-1)^{\overline{i}\cdot\overline{j}}$, wobei $\overline{i}$ die Binärdarstellung von i ist und $\overline{i} \cdot \overline{j}$ das Skalar-Produkt der zwei n-Bit langen Zeichenketten $\overline{i}$ und $\overline{j}$ ist. Dadurch wird $\overline{j}$auf $\overline{i}$ projeziert.

- Messe den Ergebniszustand. Falls $C(S_v) = 1$, so gibt es einen eindeutigen Zustand S_v, der mit einer Wahrscheinlichkeit von mindestens $\frac{1}{2}$ Endzustand des Systems ist.

Teil I

Alles verstanden?

Da der Algorithmus nicht ganz intuitiv verständlich ist, muss auf die Details noch genauer eingegangen werden. Der Schritt (1) sollte klar sein, Schritt (2) wahrscheinlich nicht. Dieser Schritt stellt den Kern des Algorithmus dar. Ziel der Iterationen ist es, wie bereits erwähnt, die Amplitude des gesuchten Zustandes zu erhöhen und die Amplituden aller nicht-gesuchten Zustände zu verringern, so daß eine Messung mit größter Wahrscheinlichkeit den gesuchten Zustand ermittelt. Die graphische Darstellung sieht man unter (2.5)

Jede Iteration in (2) erhöht die Amplitude des gesuchten Zustandes um $\frac{1}{\sqrt{N}}$. Da es $\sqrt{N}$ Iterationen über (2) gibt, erreicht die Amplitude und damit auch die Wahrscheinlichkeit 1. Es sei am Rande erwähnt, dass die Amplitude nicht monoton mit der Anzahl der Iterationen steigt. Daher reicht es auch nicht aus, schon vor der letzten Iteration die Messung durchzuführen, da zu einem Iterationspunkt vor der letzten Iteration unter Umständen ein anderer Zustand S_i mit $i \neq v$ die größte Amplitude hat.

Im folgenden wird gezeigt, dass man D als IAA (Inversion About Average) interpretieren kann und dass sie unitär ist, was für die Benutzung in einem Algorithmus für Quantencomputer erforderlich ist. Nun zunächst einmal zur Definition des IAA:

Das System befinde sich in einer Superposition

$$\sum_{x=0}^{N-1} \alpha_x \left|x\right\rangle$$

Der Mittelwert der Amplituden α definiert sich wie folgt:

$$\alpha = \frac{1}{N} \sum_{x=0}^{N-1} \alpha_x$$

Die Anwendung der IAA-Matrix A auf die Superposition invertiert die Amplituden der einzelnen Zustände am Mittelwert α:

$$A \sum_{x=0}^{N-1} \alpha_x = \sum [\alpha - (\alpha_x - \alpha)] \left|x\right\rangle$$

Nun zum eigentlichen Beweis. Zur Erinnerung: $D_{ij} = \frac{2}{N},\ falls\ i \neq j\ und\ D_{ii} = -1 + \frac{2}{N}$. D kann dargestellt werden als $D \equiv -I + 2P$, wobei I die Identitätsmatrix ist und P eine Projektionsmatrix mit $P_{ij} = \frac{1}{N}\ \forall\, i, j$. Klar ersichtlich ist, dass $P^2 = P$, darüber hinaus, dass die Anwendung von P auf einen beliebigen Vektor $\overline{v}$ einen Vektor $\overline{v'}$ erzeugt, bei dem jede Komponente gleich dem Durchschnittswert aller Komponenten ist. Aus $P^2 = P$ folgt: Wenn $D \equiv -I + 2P$, dann ist $D^2 = I$ und damit auch unitär. Nun wenden wir D auf einen beliebigen Vektor $\overline{v}$ an. Es ergibt sich $D\overline{v} = (-I + 2P)\overline{v} = \overline{v} + 2P\overline{v}$. Aus obiger Diskussion folgt, dass jede Komponente des Vektors $P\overline{v}$ A ist, wobei A das Mittel aller Komponenten des Vektors $\overline{v}$ ist. Deswegen ist die i-te Komponente des Vektor $D\overline{v}$ durch $(-v_i + 2A)$ gegeben. Man kann diesen Ausdruck auch als $(A + (A - v_i))$ schreiben. Wenn man nun die IAA auf einen Vektor anwendet, bei dem alle ausser eine Komponente einen Wert $C \approx \frac{1}{\sqrt{N}}$ haben (die einzige andere Komponente soll einen negativen Wert haben), dann ist das Mittel A in etwa gleich C. Da jede der $(N-1)$ Komponenten ungefähr gleich dem Mittel A ist, ändert sich die Amplitude und damit die Wahrscheinlichkeit nicht signifikant durch die IAA. Die eine ursprünglich negative Komponente wird dann jedoch positiv und ihr Betrag steigt um etwa $2C$ an, was in etwa $\frac{2}{\sqrt{N}}$ ist.

Zu beweisen ist, dass die Amplitude durch jede Iteration um $\frac{1}{\sqrt{N}}$ ansteigt.

Theorem 1

Behauptung: Die Diffusions-Transformierung kann man durch $D = WRW$ implementieren, wobei W die Walsh-Hadamard-Transformation und R die Rotationsmatrix wie folgt definiert sind:

$$\begin{array}{ccccc} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & -1 \end{array}$$

Beweis: Wie bereits bekannt, ist $W_{ij} = 2^{-\frac{n}{2}}(-1)^{\bar{i}\cdot\bar{j}}$, wobei $\bar{i}$ die Binärdarstellung von i ist und $\bar{i} \cdot \bar{j}$ das Skalar-Produkt der zwei n-Bit langen Zeichenketten $\bar{i}$ und $\bar{j}$ ist. R kann man auch als $R = R_1 + R_2$ schreiben, wobei $R_1 = -I$ (I ist die Identitätsmatrix) und $R_{2,\,00} = 2$, $R_{2,\,ij} = 0$ wenn $i \neq 0,\ j \neq 0$. Da $MM = I$ (M ist die Hadamard-Matrix, siehe oben) ist gezeigt, daß auch $WW = I$ und deshalb gilt $D_1 = WR_1W = -I$. Durch Matrix-Multiplikation ergibt sich $D_{2,\,ad} = \Sigma_{bc} W_{ab} R_{2,\,bc} W_{cd}$. Aus der Definition von R_2 und $N = 2^n$ folgt $D_{2,\,ad} = 2W_{a0}W_{0d} = \frac{2}{2^n}(-1)^{\bar{a}\cdot\bar{0}+\bar{0}\cdot\bar{d}} = \frac{2}{N}$. So sind alle Elemente der Matrix $D_2 = \frac{2}{N}$, die Summe der Matrizen D_1 und D_2 ergibt D . □

Korrolar 1

Behauptung: Habe der Zustandsvektor für einen (und nur einen) beliebigen Zustand die Amplitude k und für jeden der $(N-1)$ übrigen Zustände die Amplitude l . Seien k und l reelle Zahlen, $k < 0$ und $l > 0$ und $\left|\frac{k}{l}\right| < \sqrt{N}$. Nach Anwendung der Diffusions-Transformation sind k und l positiv.

Beweis: Aus Theorem 2 ist bekannt, daß

$$k_1 = \left(\frac{2}{N} - 1\right) k + 2\frac{(N-1)}{N} l$$

Unter der Annahme, daß $N > 2$ sei, folgt daß $\left(\frac{2}{N} - 1\right) < 0$ ist. Nach Vorraussetzung gilt $k < 0$ und $2\left(\frac{(N-1)}{N}\right) l > 0$ und deswegen ist auch $k_1 > 0$. Ähnlich dazu folgt auch aus Theorem 2, daß $l_1 = \frac{2}{N}k + \frac{(N-2)}{N} l$ gilt, und wenn die Bedingung $\left|\frac{k}{l}\right| < \frac{(N-2)}{2}$ erfüllt ist, dann gilt auch $l_1 > 0$. Wenn $\left|\frac{k}{l}\right| < \sqrt{N}$ ist, dann ist für $N \geq 9$ die Bedingung $\left|\frac{k}{l}\right| < \frac{(N-2)}{2}$ erfüllt und damit ist $l_1 > 0$. □

Theorem 2

Behauptung: Habe der Zustandsvektor für einen (und nur einen) beliebigen Zustand die Amplitude k_1 und für jeden der $(N-1)$ übrigen Zustände die Amplitude l_1 . Nach Anwendung der Diffusions-Transformation D ist die Amplitude in dem einen Zustand $k_2 = \left(\frac{2}{N} - 1\right) k_1 + 2\frac{(N-1)}{N} l_1$ und in allen $(N-1)$ übrigen Zuständen ist die Amplitude $l_2 = \frac{2}{N} k_1 + \frac{(N-2)}{N} l_1$.

Beweis: Nach der Definition der Diffusions-Transformation gilt:

$$k_2 = \left(\frac{2}{N} - 1\right) k_1 + 2\frac{(N-1)}{N} l_1$$

$$l_2 = \left(\frac{2}{N} - 1\right) l_1 + \frac{2}{N} k_1 + \frac{2(N-2)}{N} l_1$$

Deswegen ist $l_2 = \frac{2}{N} k_1 + \frac{(N-2)}{N} l_1$ □

Korrolar 2

Behauptung: Der Zustandsvektor sei wie folgt beschaffen: für $C(S) = 1$ sei die Amplitude k , für jeden der $(N-1)$ übrigen Zustände sei die Amplitude l . Nach Anwendung der Diffusions-Transformation D sind die Amplituden k_1 bzw. l_1 wie aus Theorem 2 bekannt: $k_1^2 + (N-1)l_1^2 = k^2 + (N-1)l^2$.

Beweis: Aus Theorem 2 folgt, daß

$$k_1^2 = \frac{(N-2)^2}{N^2} k^2 + 4\frac{(N-1)^2}{N^2} l^2 - \frac{4(N-2)(N-1)}{N^2} kl$$

ist. Ähnlich gilt:

$$(N-1)l_1^2 = \frac{4(N-1)^2}{N^2} k^2 + \frac{(N-2)^2}{N^2}(N-1)l^2 + \frac{4(N-2)(N-1)}{N^2} kl$$

Aus der Addition der beiden Gleichungen folgt die Behauptung. □

Theorem 3

Behauptung: Sei der Zustandsvektor vor Stufe (2.)(a) des Algorithmus folgender: für den Zustand $C(S) = 1$ sei die Amplitude k , für alle anderen $(N-1)$ Zustände sei die Amplitude l , so daß $(o < k < \frac{1}{\sqrt{2}})$ und $l > 0$. Die Änderung Δk nach den Schritten (a) und (b) des Algorithmus hat eine untere Schranke von $\Delta k > \frac{1}{2\sqrt{N}}$. $l > 0$ ist per Behauptung gewährleistet.

Beweis: Die initialen Amplituden seien mit k und l , die Amplituden nach der Phaseninversion (bei Schritt (a)) mit k_1 und l_1 sowie nach der Diffusions-Transformation (bei Schritt (b)) mit k_2 und l_2 gekennzeichnet. Aus Theorem 2 folgt, dass:

$$k_2 = (1 - \frac{2}{N})k + 2(1 - \frac{1}{N})l$$

. Deswegen gilt: $\Delta k = k_2 - k = -\frac{2k}{N} + (1 - \frac{1}{N})l$

Da $(0 < k < \frac{1}{\sqrt{2}})$, folgt aus dem Korrolar 2, daß $|l| > \frac{1}{\sqrt{2N}}$ und aus der Behauptung des Theorems l sei positiv, folgt, daß $l > \frac{1}{\sqrt{2N}}$. Unter Annahme eines nicht-trivialen N folgt, daß $\Delta k > \frac{1}{2\sqrt{N}}$ ist. Um zu beweisen, daß $l_2 > 0$, betrachte man, daß nach der Phaseninversion (bei Schritt (a)) $k_1 < 0$ und $l_1 > 0$ ist. Desweiteren folgt aus $(0 < k < \frac{1}{\sqrt{2}})$ und $(l) > \frac{1}{\sqrt{2N}}$ direkt $\left|\frac{k_1}{l_1}\right| < \sqrt{N}$. Deswegen ist nach Korrolar 1 l_2 positiv. □

Aus diesem Theorem folgt unmittelbar, daß es eine Zahl $M < \sqrt{2N}$ gibt, so daß in M Iterationen von Schritt (ii) $k > \frac{1}{\sqrt{2}}$. Da die Wahrscheinlichkeit des Systems in einem beliebigen Zustand gefunden zu werden proportional zum Quadrat der Amplitude ist, folgt daraus, daß das System genau dann im gewünschten Zustand ist (C(S)=1), wenn $k = \frac{1}{\sqrt{2}}$, $k^2 = \frac{1}{2}$ ist. Wenn das System nun gemessen wird, wird es sich mit einer Wahrscheinlichkeit von mehr als $\frac{1}{2}$ im gewünschten Zustand befinden.

Teil II
Diskussion des Algorithmus

Wie wir gesehen haben, stellt Grover's Algorithmus eine verglichen mit klassischen Algorithmen sehr effiziente Lösung der Datenbanksuche dar. Im folgenden wird noch eine Aufwandsbetrachtung des Algorithmus durchgeführt und eine technische Implementierung gezeigt.

2.1 Verschränkung

Grover's Algorithmus nutzt verschränkte Zustände (Beispiel: Zustand nach Phaseninversion bei einem 2-QuBit-System: $|00\rangle - |01\rangle + |10\rangle + |11\rangle$ ist verschränkt, da er nicht weiter faktorisierbar ist). Diese Tatsache erschwert technische Umsetzungen, allerdings hat Meyer [3] nachgewiesen, dass man jeden (!) Quantenalgorithmus auch ohne verschränkte Zustände implementieren kann. Wenn man die Struktur des Tensorproduktes im Hilbert'schen Raum ausser Acht lässt und ein

N-Zustands-Orakel anstelle eine n-QuBit-Orakels benutzt, existieren per Definition keine verschränkten Zustände mehr. Dies ließe sich experimentell einfacher, z.B. als Atom umsetzen. Die 2^n Zustände des Atoms wären dann durch Rydbergzustände gewährleistet. Damit ergibt sich aber das Problem exponentiell steigenden Implementierungsaufwandes.

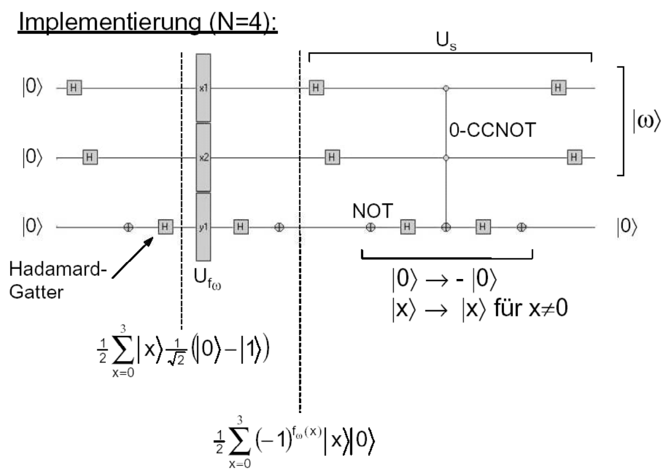
2.2 Quantenalgorithmen in Makroskopischer Umgebung

Obwohl die Vermutung nahe liegt, Quantenalgorithmen wären nur in der Quantenwelt experimentell umsetzbar, so zeigt das folgende - ebenfalls von Grover [2] vorgestellte - Problem eine erstaunliche Lösung:

Gegeben Sei ein System von N klassischen Pendeln, wobei alle bis auf eines identisch sind, d.h. sie haben alle dieselbe Pendellänge. Weiterhin ist die Pendellänge des einen Pendels nur wenig von der Länge der anderen Pendel unterschiedlich und bekannt. Die Aufgabe ist es, die Position des einen Pendels zu finden.

Die Lösung basiert auf dem Prinzip der Amplitudenverstärkung. Durch eine geschickte Kopplung der Pendel aneinander und durch Energieübertrag durch ein zusätzliches Pendel konnte Grover zeigen, daß man auch hier nur etwa $\sqrt{N}$ Schritte benötigt, um das eine Pendel zu finden.

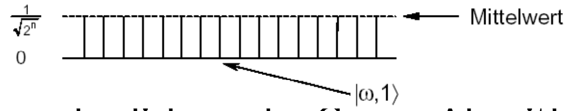
2.3 Technische Umsetzung



$U_s := 2\,|s\rangle\langle s| - \mathbf{1}$, $|s\rangle = H^{(n)}\,|0\rangle \Rightarrow U_s = H^{(n)}\,(2\,|0\rangle\langle 0| - \mathbf{1})\,H^{(n)}$

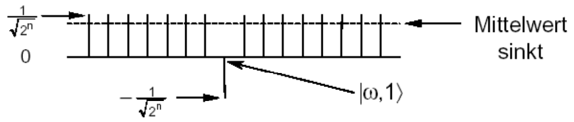
$H^{(n)}$: bitweise Hadamard-Transformation

Amplituden der (Basis-)Zustände $|x_i, f(x_i)\rangle$ nach Schritt 1:

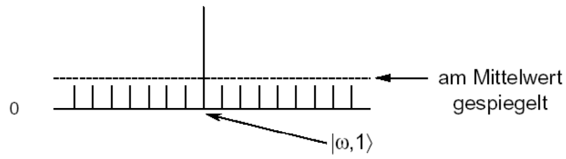


2.4 Veranschaulichung des Grover Algorithmus

nach einem „speziellen *Vorzeichenwechsel*“:



nach „*Inversion about the average*“:



Literatur

[1] P. W. Shor: Algorithms for quantum computation: diescrete logarithms and factoring, Proceedings, 35th Annual Symposium on Fundamentals of Comp. Science (FOCS), 1994, Seiten 116-123

[2] L.K. Grover, A.M. Sengputa: A CLASSICAL ANALOG OF QUANTUM SEARCH, quant-ph/0109123

[3] David A. Meyer: Sophisticated Quantum Search Without Entanglement; Physical Review Letters, Vol. 85, No. 9