Quantenalgorithmen
Suchen mit nur einer Anfrage

Frank-Michael Krupp
[http://www.vuni.de]

Seminar im Sommersemester 2000
Universität Karlsruhe
Institut für Algorithmen und Kognitive Systeme (IAKS)
Betreuer: Dipl. Informatiker Markus Grassl
Juli 2000

Inhaltsverzeichnis

1 Einleitung

Der Physik-Nobelpreisträger Richard Feynman entwickelte 1982 als einer der ersten die Idee vom Computer, der mit Quanteneffekten Informationen verarbeitet. Ein klassisches Teilchen kann sich nur in einem bestimmten Zustand (0 oder 1) befinden. In der Quantenphysik kann aber ein Teilchen in einem Zwischenzustand zwischen mehreren Zuständen sein, der so genannten Superposition. Mit einem 8 Quantenbit(Qubit)-Register kann der Rechner in 2^8 Basiszuständen sein. Jede unitäre[1] Operation wirkt sich auf alle Basiszustände gleichzeitig aus. Somit entsteht ein massiver Parallelrechner. Die Messung dieser Superposition führt jedoch dazu, dass die Teilchen wieder einen eindeutigen Zustand annehmen (Dekohärenz), $2^8 - 1$ Zustände gehen unwiederbringlich verloren. Welcher Zustand weiterbesteht, kann man mit dem Manipulieren der Wahrscheinlichkeitsamplitude beeinflussen. Nach und nach entstanden spezielle Algorithmen, die diese Besonderheiten ausnutzen konnten, und deshalb statt exponentiellem nur noch polynomialen Aufwand benötigen.
Der Durchbruch gelang Peter Shor 1994 mit einem Faktorisierungsalgorithmus der statt exponentiellem Aufwand[2] nur noch linearen Aufwand braucht.[SH94]
Der Suchalgorithmus von Lov Grover (Bell Laboratories), mit dem eine nicht strukturierte Datenbank, z. B. Telefonbuch anhand bekannter Telefonnummer, bestehend aus N Elementen in $\sqrt{N/4}$ Schritten und nicht wie bisher klassisch in $N/2$ Schritten durchsucht werden kann, war der erste bedeutende Algorithmus mit dem man beweisen konnte, dass Quantenalgorithmen den klassischen Algorithmen überlegen sind. Mit dem hier vorgestellten Algorithmus, von Grover 1997 entwickelt, gelingt die Suche in O(1), also konstant vielen Anfragen. Er ist eine Abwandlung von Grovers ursprünglichen Suchalgorithmus, deshalb gehe ich im folgenden Abschnitt kurz auf diesen ein. Zu einem ähnlichen Resultat kamen Terhal und Smolin [TS98] mit einem anderen Ansatz.

[1] $A * A^* = I$; alle unitären Transformationen erfüllen die Anforderung, reversibel zu sein

[2] Aufwand des bisher schnellsten klassischen Algorithmus (Number field sieve): $O(e^{(\log n)^{\frac{1}{3}} (\log \log n)^{\frac{2}{3}}})$

2 Motivation des Algorithmus

2.1 Standard-Grover-Suchalgorithmus

Mit Grovers Algorithmus ist es möglich, aus einer Datenbank von $N = 2^n$ Einträgen ein markiertes Element x_0 mit $O(\sqrt{N})$ Quantenanfragen zu finden. Dies ist immer noch ein exponentieller Aufwand um „die Nadel im Heuhaufen“ zu finden, leider wird es bis auf konstante Faktoren auch nicht schneller gehen.
Grovers Algorithmus funktioniert kurzgefaßt so:

1. Erzeuge mit n Quantenbits eine Superposition aus $2^n = N$ Zuständen. Ein Basiszustand bezieht sich auf ein Element der Datenbank.
2. Invertiere die Phase des gesuchten Zustandes.
3. Führe die Inversion-About-Average-Transformation durch, die die Wahrscheinlichkeitsamplitude des Zustandes x_0 anhebt.
4. Wiederhole Schritt 2 und 3 $O(\sqrt{N})$-mal[3], man erhält bei der anschließenden Messung des Quantensystems mit hoher Wahrscheinlichkeit den gesuchten Wert x_0.

2.2 Erster Ansatz

Wie kann man die Wiederholung der Schritte 2 und 3 einsparen? Eine Möglichkeit ist, die Anfragen zu parallelisieren, vgl. Abb. 1. Der naive Ansatz besteht darin, viele Teilsysteme zu initialisieren, jeweils einmal per Datenbankanfrage die Phase des gesuchten Zustands zu invertieren und anzuheben und dann ohne weitere Wiederholungen sofort zu messen. Bei dieser Messung erhält man in jedem Teilsystem den gesuchten Zustand mit einer etwas höheren Wahrscheinlichkeit als einen falschen Zustand. Eine Mehrheitsentscheidung unter den gemessenen Zuständen liefert das gewünschte Ergebnis. Damit diese Mehrheitsentscheidung das richtige Ergebnis liefert, müssen sehr viele Teilsysteme vorhanden sein, und da jedes eine Datenbankanfrage durchführt, ist mit diesem ersten Ansatz nichts gewonnen.

2.3 Codierungstrick

Die Idee mit den Teilsystemen und der Mehrheitsentscheidung führte Grover auf eine weitere Verbesserungsmöglichkeit. Durch eine geschickte Codierung der Anfragen läßt sich ein Anfragevektor bilden.

[3]Nicht häufiger, da dann an dem gewünschten Zustand vorbeigedreht wird. Die Wahrscheinlichkeit sinkt bei jeder Wiederholung.

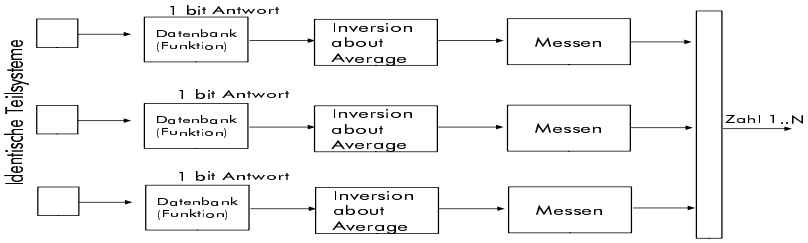

Abbildung 1: erster Ansatz für Parallelisierung

Dieser wird an die Datenbank geleitet und mit der Antwort und einer entsprechend trickreichen Codierung ist es möglich, wieder auf die einzelnen Teilsysteme zurückzuschließen (vgl. Abb. 2).

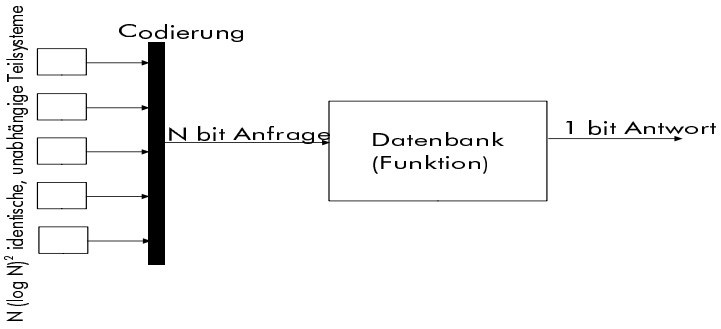

Abbildung 2: Parallelisierung mit geschickter Codierung

3 Ablauf des Algorithmus

3.1 Spezifikation des Problems

Die Aufgabe besteht darin, ein markiertes Element in einer nicht strukturierten Datenbank von $N = 2^n$ Elementen $A_1, \ldots, A_N = M$ zu suchen. Dabei sollen möglichst wenig Anfragen ausreichen. Alle Antworten bestehen aus 1 Bit, sprich ja/nein.
Andere Sichtweise: Es ist eine Funktion $F(x)$ gegeben, finde ein x_0, für das die Funktion den Wert 1 annimmt mit möglichst wenig

Funktionsauswertungen, d. h., zu $F : M \to \{0, 1\}$ finde $x_0 \in M$ mit $F(x_0) = 1$, wobei $F(x) = 0\ \forall x \neq x_0$
Im folgenden gebe es nur einen gesuchten Zustand.

3.2 Initialisierung

Erzeuge eine große Anzahl η identischer Teilsysteme ($\eta = N(\log N)^2$, Erklärung in 3.7). Jedes System besteht aus ν Qubits, es können also $2^\nu = N$ verschiedene Zustände eingenommen werden. N ist zugleich die Anzahl der Elemente in der zu durchsuchenden Datenbank, es besteht eine Bijektion zwischen Zuständen und Datenbankelementen. Der Gesamtzustand ist wie folgt:

$$\begin{array}{ccccccc} |\psi_1\rangle & = & \underbrace{(|0\rangle\,|0\rangle \ldots |0\rangle)}_{\nu\eta} & \otimes & \underbrace{(|0\rangle\,|0\rangle \ldots |0\rangle)}_{N} & \otimes & \underbrace{|0\rangle}_{1} \\ \text{Zustand1} & & \eta \text{ Teilsysteme} & & & & \text{Anfrage} \end{array}$$

Initialisiere die Teilsysteme in einer Superposition. Dazu wird üblicherweise die Hadamard-(Fourier 2)-Transformation verwendet. Dies ist die unitäre Matrix $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Das Quantenbit ($|0\rangle$, das anfangs mit der Wahrscheinlichkeit $p = 1$ bei einer späteren Messung den Wert 0 liefert, wird nun in den Zwischenzustand $\frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle)$ gesetzt ($\frac{1}{\sqrt{2}}$ sorgt für die Normierung). Bei einer Messung wird dieser Zustand zerstört und man erhält mit $p = \frac{1}{2}$ den Wert 0 und mit $p = \frac{1}{2}$ den Wert 1. Diese Hadamard-Transformation wird auf die ersten $\eta\nu$ Bits und das letzte Bit angewendet und man erhält den Zustand

$$\left((\underbrace{H \otimes H \otimes \ldots \otimes H}_{\nu\eta}) \otimes \left(\underbrace{I \otimes I \otimes \ldots \otimes I}_{N} \right) \otimes H \right) |\psi_1\rangle$$

$$\begin{array}{cccccc} = & (H\,|0\rangle\,|0\rangle) & \otimes & (I\,|0\rangle\,|0\rangle) & \otimes & (H\,|0\rangle) \\ = & ((|0\rangle + |1\rangle) \otimes \ldots \otimes (|0\rangle + |1\rangle)) & \otimes & (0 \otimes \ldots \otimes 0) & \otimes & (|0\rangle + |1\rangle) \end{array}$$

(Bem.: Der Normierungsfaktor $\frac{1}{\sqrt{2^{\nu\eta+1}}}$ fehlt hier und wird auch im weiteren nicht mehr mitgeführt).
Damit hat man nun in den η Teilsystemen die gleichen Wahrscheinlichkeitsamplituden $\left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}}\right)$. Rotiere die Phase des letzten Quantenbits mit der Transformation $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, dies

invertiert die Phase des Quantenbit:

$$\begin{aligned} Z\left(|0\rangle + |1\rangle\right) &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} * \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] \\ &= \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right] \\ &= \left(|0\rangle - |1\rangle\right) \end{aligned}$$

Zusammenfassend ist das Quantenregister nach diesem Initialisierungsschritt in folgendem Zustand:

$$|\psi_2\rangle = \underbrace{\left(\left(|0\rangle + |1\rangle\right) \otimes \ldots \otimes \left(|0\rangle + |1\rangle\right)\right)}_{\begin{array}{l}\text{Kurzschreibweise } \left(|0\rangle + |1\rangle\right)^{\otimes \nu\eta} \\ \text{oder } \left(\sum_{x=1}^{N} |x\rangle\right)^{\otimes \eta}\end{array}} \otimes \; |0\rangle^{\otimes N} \; \otimes \; \left(|0\rangle - |1\rangle\right)$$

3.3 Codierung der Anfrage

Nun folgt der Trick des Algorithmus. Die Information der $\nu\eta$ Quantenbits wird auf einen N Bit großen Indikatorvektor $(\chi_1, \chi_2, \chi_3, \ldots, \chi_N)$ komprimiert. Durch die spezielle Wahl dieser Codierung wird es in einem späteren Schritt möglich sein, die Antwort wieder auf das gesamte System anzuwenden. Dieser Indikatorvektor berechnet sich folgendermaßen: Jedes Teilsystem ist unabhängig von den anderen und bezieht sich in der Basisdarstellung auf eines der $1, \ldots, N$ Datenbankelemente. Zähle die Teilsysteme, die mit dem Element j korrespondieren, prüfe, ob diese Zahl gerade ist und speichere die Parität in χ_j.

$$\begin{array}{lll} j = 1, \ldots, N \quad \chi_j : & \{0,1\}^{\nu\eta} & \rightarrow \quad \{0,1\} \\ & \left(\sum_{x=1}^{N} |x\rangle\right)^{\otimes \eta} & \mapsto \quad \begin{cases} 1 \text{ falls Anzahl der } x_i \text{ mit} \\ \text{ Wert } j \text{ ungerade} \\ 0 \quad \text{sonst} \end{cases} \end{array}$$

Diese zwischengespeicherten N Werte (0 oder 1) stellen die Anfrage an die Datenbank dar. Diese Bits werden an das Datenbankorakel gegeben, das dann die Antwort 0 oder 1 gibt. Weitere Anfragen sind nicht erforderlich. Mit diesem 1 Bit Information der Antwort wird es möglich sein, die zu dem gesuchten Zustand x_0 gehörenden Wahrscheinlichkeitsamplituden zu erhöhen.

Der Zustand nach diesem Schritt ist:

$$|\psi_3\rangle = \sum_{\mathrm{x=1}}^{\mathrm{N}} |x_1\rangle\,|x_2\rangle \ldots |x_\eta\rangle \quad \otimes \quad (|\chi_1\rangle \otimes \ldots \otimes |\chi_N\rangle) \quad \otimes \quad (|0\rangle - |1\rangle)$$

3.4 Verarbeitung der Anfrage

An die Datenbank wird der Indikatorvektor $(\chi_1, \ldots, \chi_N)$ übergeben. Wenn der zu x_0 gehörende Indikator χ_{x_0} den Wert 1 hat, dann ist die Antwort 1, ansonsten 0.

$$P(\underbrace{\chi_1, \ldots, \chi_N}_{\mathrm{Nbit}}) \rightarrow \underbrace{\chi_{x_0}}_{\mathrm{1bit}}$$

Betrachte dann die folgende Transformation

$$U_P : \left| \left(\sum_{\mathrm{x=1}}^{\mathrm{N}} |x\rangle \right), (|0\rangle - |1\rangle) \right\rangle \rightarrow \left| \left(\sum_{\mathrm{x=1}}^{\mathrm{N}} |x\rangle \right), (|0\rangle - |1\rangle) \oplus P(x) \right\rangle$$

auf *ein* Teilsystem angewendet. Diese unitäre Transformation U_P wird in allen Zuständen, die sich auf x_0 beziehen, die Amplitude ändern und die übrigen Zustände und das hintere Hilfsbit unverändert lassen. Warum das so ist, ist ersichtlich, wenn man die Zustände in die zwei Mengen $A = \{x|P(x) = 0\}$ und B=$\{x|P(x) = 1\}$ aufteilt.

$$\begin{aligned}
&U_P\left(|\psi, (|0\rangle - |1\rangle)\rangle\right) \\
&= U_P\left(\sum_x |x\rangle \otimes (|0\rangle - |1\rangle)\right) \\
&= U_P\left(\left(\sum_{x\in A} |x\rangle + \sum_{x\in B} |x\rangle\right) \otimes (|0\rangle - |1\rangle)\right) \\
&= U_P\left(\sum_{x\in A} |x,0\rangle - \sum_{x\in A} |x,1\rangle + \sum_{x\in B} |x,0\rangle - \sum_{x\in B} |x,1\rangle\right) \\
&= \left(\sum_{x\in A} |x,0\oplus 0\rangle - \sum_{x\in A} |x,1\oplus 0\rangle + \sum_{x\in B} |x,0\oplus 1\rangle - \sum_{x\in B} |x,1\oplus 1\rangle\right) \\
&= \left(\sum_{x\in A} |x,0\rangle - \sum_{x\in A} |x,1\rangle + \sum_{x\in B} |x,1\rangle - \sum_{x\in B} |x,0\rangle\right) \\
&= \left(\left(\sum_{x\in A} |x\rangle - \sum_{x\in B} |x\rangle\right) \otimes (|0\rangle - |1\rangle)\right)
\end{aligned}$$

Der $\oplus$-Operator ist XOR oder die bitweise Addition modulo 2. Entscheidend ist, dass sich der Faktor $(-1)^{P(x)}$ in Tensorprodukten nach vorne ziehen läßt und sich auf die einzelnen Teilsysteme in der

Form $(-1)^{f(x)}$ verteilen läßt. Diese Umformung ist nicht offensichtlich, der genaue mathematische Beweis ist in [GB97] zu finden, hier ein vereinfachender Erklärungsversuch: Ist die Anzahl der Teilsysteme, die sich auf das x_0 beziehen, gerade, so ergibt die Datenbankanfrage die 0 und damit $(-1)^0 = 1$. Es kann eine gerade Anzahl von 1 auf die Teilsysteme in der Form $(-1)^{f(x)}$ vor die betreffenden Systeme verteilt werden ($(-1)^{2n} = 1$). Ist die Anzahl dagegen ungerade, dann ist $(-1)^1 = -1 = (-1)^{2n+1} = (-1)^{f(x)}$.
Damit ist das Quantensystem im Zustand

$$|\psi_4\rangle = \left(\sum_{\mathrm{x}=1}^{\mathrm{N}} (-1)^{f(x)} |x\rangle \right)^{\otimes \eta} \otimes (|0\rangle)^{\otimes N} \otimes (|0\rangle - |1\rangle)$$

3.5 Inversion-About-Average

Nun kommt der anfangs erwähnte Inversion-About-Average-Operator zum Zug, der wie folgt definiert ist

$$D = \begin{pmatrix} -1+\frac{2}{N} & \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & -1+\frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} & -1+\frac{2}{N} & \dots & \frac{2}{N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \frac{2}{N} & \dots & -1+\frac{2}{N} \end{pmatrix}$$

Der Operator D ist unitär, $DD^* = I$ folgt aus einfachem Nachrechnen. Wende D auf jedes Teilsystem an:

$$D^{\otimes \eta} = \underbrace{D \otimes D \otimes \ldots \otimes D}_{\eta - mal}$$

Jedes Teilsystem hat den Wahrscheinlichkeitsvektor

$$\left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}} \ldots, \frac{1}{\sqrt{N}}, \underbrace{-\frac{1}{\sqrt{N}}}_{\text{an Stelle } x_0}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}} \right)$$

D darauf angewendet ergibt:

$$\begin{pmatrix} -1+\frac{2}{N} & \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & -1+\frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} & -1+\frac{2}{N} & \dots & \frac{2}{N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \frac{2}{N} & \dots & -1+\frac{2}{N} \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{\sqrt{N}} \\ \frac{1}{\sqrt{N}} \\ \dots \\ \frac{1}{\sqrt{N}} \\ -\frac{1}{\sqrt{N}} \\ \frac{1}{\sqrt{N}} \\ \dots \\ \frac{1}{\sqrt{N}} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{\sqrt{N}}\left(-1+\frac{2}{N}+(N-3)\frac{2}{N}\right) \\ \frac{1}{\sqrt{N}}\left(-1+\frac{2}{N}+(N-3)\frac{2}{N}\right) \\ \dots \\ \frac{1}{\sqrt{N}}\left(-1+\frac{2}{N}+(N-3)\frac{2}{N}\right) \\ \frac{1}{\sqrt{N}}\left(1-\frac{2}{N}+(N-1)\frac{2}{N}\right) \\ \frac{1}{\sqrt{N}}\left(-1+\frac{2}{N}+(N-3)\frac{2}{N}\right) \\ \dots \\ \frac{1}{\sqrt{N}}\left(-1+\frac{2}{N}+(N-3)\frac{2}{N}\right) \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{N}}(1-\frac{4}{N}) \\ \frac{1}{\sqrt{N}}(1-\frac{4}{N}) \\ \dots \\ \frac{1}{\sqrt{N}}(1-\frac{4}{N}) \\ \frac{3}{\sqrt{N}} \\ \frac{1}{\sqrt{N}}(1-\frac{4}{N}) \\ \dots \\ \frac{1}{\sqrt{N}}(1-\frac{4}{N}) \end{pmatrix}$$

Für großes N ist $4/N$ nicht relevant und die Wahrscheinlichkeit des Zustandes x_0 steigt auf das Neunfache der übrigen.
Diese Transformation kann auch anhand der Abbildungen 3 - 5 veranschaulicht werden.

Sei $E = \frac{1}{\sqrt{N}}$. Der Zustand x_0 wird invertiert und hat den Wert $-1E$. Die Inversion-About-Average spiegelt den Zustand an der durch den Durchschnitt der Amplituden gegebenen Achse. Da der Zustand um $-2E$ unter dieser Achse lag, ist er danach auf $1E - (-2E) = 3E$ gestiegen.
Zwischenbemerkungen:

1. Im Standard-Grover-Suchalgorithmus werden diese Operationen nun $O(\sqrt{N})$-mal wiederholt, dann ist die Wahrscheinlichkeit hinreichend groß.

2. Diese Inversion-About-Average und somit der gesamte Algorithmus funktioniert auch mit mehreren ausgewählten Zuständen, allerdings darf die Anzahl nicht $N/4$ übersteigen, da dann nicht mehr zwischen ausgewählten Zuständen und nicht ausgewählten differenziert werden kann.

3.6 Auswertung

Im letzten Schritt wird jedes der η Teilsysteme gemessen. Dadurch geht durch Dekohärenz die Superposition verloren und man erhält in

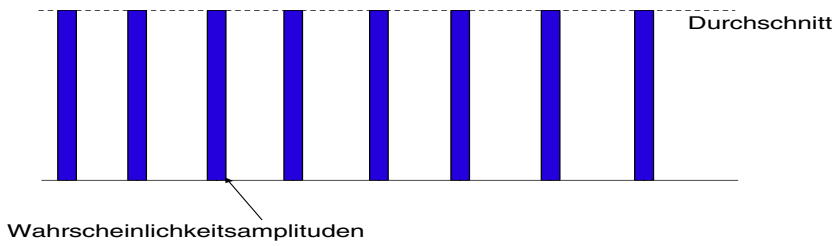

Abbildung 3: Anfangsinitialisierung des Systems mit Wahrscheinlichkeitsvektor $(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}})$

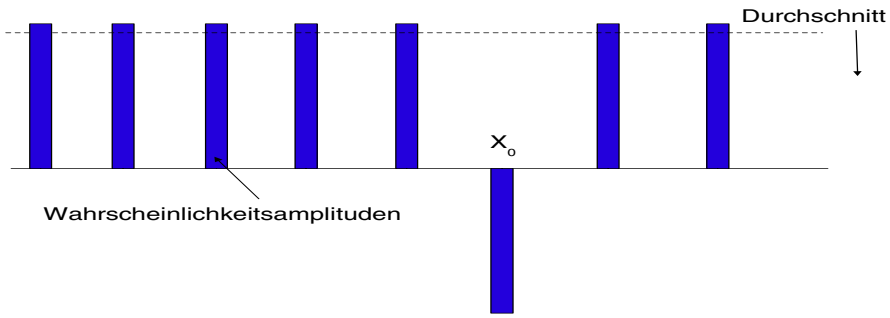

Abbildung 4: Zustand nach Inversion
$(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}}, -\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}})$

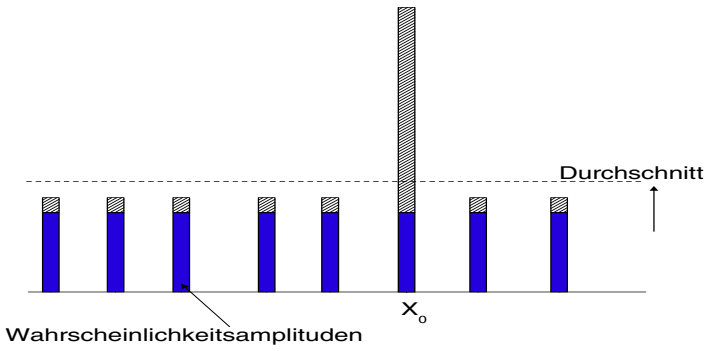

Abbildung 5: Wahrscheinlichkeit anheben
$(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}}, 3\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}})$

jedem Teilsystem eine Zahl zwischen 1 und N, die auf das entsprechende Datenbankelement verweist. Wie gesehen ist die Wahrscheinlichkeit, dass bei den Systemen der richtige Zustand nicht zerstört wird, etwas höher als bei den übrigen. Intuitiv ist klar, dass das am häufigsten ausgewählte Element das gesuchte Element darstellt. Eine geschlossene Formel läßt sich nur schwer angeben. Gesucht ist die Wahrscheinlichkeit für das Ereignis „x_0 wird K-mal gemessen“ unter der Bedingung, dass alle anderen Zustände weniger als K-mal gemessen werden (Binomialverteilung mit Parametern Anzahl η und Erfolgswahrscheinlichkeit $\frac{9}{N}$).
Für die Anzahl der Teilsysteme gilt: Es werden $\eta = N \log(N)$ Teilsysteme benötigt.
Erklärung:
Die Wahrscheinlichkeit, den gewünschten Zustand zu erhalten, ist $\left(\frac{3}{\sqrt{N}}\right)^2 = \frac{9}{N}$, die Wahrscheinlichkeit für einen falschen Zustand ist ungefähr $\frac{1}{N}$. Hat man nun ν Teilsysteme folgt aus dem Gesetz der großen Zahlen, dass die Wahrscheinlichkeiten gegeben sind durch

$$\underbrace{\frac{9\eta}{N}}_{\text{1Zustand}} \qquad \underbrace{\frac{1\eta}{N}}_{\text{(N - 1) Zustände}}$$

Mit $\nu = KN$ erhält man $9K$ und $1K$. Bei großem K sieht man, dass die Wahrscheinlichkeit den richtigen Zustand zu messen größer ist, als einen der $N - 1$ falschen.
Nach dem zentralen Grenzwerttheorem reicht die Wahl $K = (\log(N))^2$ aus. Es sind somit $N(\log(N))^2$ Teilsysteme zu verwenden. Eine genauere Herleitung findet sich in [Gro97].

3.7 Analyse

Der Algorithmus sieht sehr viel versprechend aus, die Einsparung der Anfragen geht aber einher mit einem großen Anstieg an Vorbereitungs- und Auswertungsschritten. Das Finden durch eine Anfrage hat also, solange es nicht gelingt, den Extra-Aufwand zu vermindern, nur theoretischen Wert, es sei denn, die Anfragen an eine Datenbank kosten viel Zeit.

4 Beispiel

Zum Schluß ein Beispiel, das den Ablauf des Algorithmus anhand einer Datenbank mit $2^3 = 8$ Elementen A, B, C, D, E, F, G, H verdeutlichen soll.

Es sei F das gesuchte Element. Es ist schon bei der geringen Größe der Datenbank der große Vorbereitungsaufwand des Systems zu erkennen. Es sind nämlich $8(\log 8)^2 = 72$ Teilsysteme zu initialisieren. Jedes besteht aus 3 Quantenbits und wird mit der Hadamard-Transformation in eine Superposition gebracht. Die Wahrscheinlichkeitsamplitude ist dann:

$$\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$$

Betrachte die Teilsysteme in der Basisdarstellung und stelle fest, worauf sich die einzelnen Teilsysteme beziehen.
Man erhält beispielsweise für die 72 Teilsysteme die Verteilung

A	B	C	D	E	$\boxed{\text{F}}$	G	H
9	5	7	16	13	12	3	7

und damit den Indikatorvektor

χ_A	χ_B	χ_C	χ_D	χ_E	χ_F	χ_G	χ_H
1	1	1	0	1	$\boxed{0}$	1	1

Diese 8-Bit-Anfrage wird an das Orakel geleitet. Dieses sieht die 0 an der gesuchten Stelle, und gibt gemäß der Funktion $P(x)$ die 0 zurück. (Wäre die Wertigkeit von F gleich 13, so wäre die Parität 1 und das Orakel würde 1 zurückgeben). Gemäß der in 3.5 angegebenen Umformung wird diese $(-1)^0$ nach vorne in die Teilsysteme gezogen und die Phase des Zustandes x_0 in allen 72 Teilsystemen gedreht.

$$\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, -\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$$

Darauf die Inversion-About-Average-Transformation angewendet ergibt:

$$\begin{pmatrix}
-\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4}
\end{pmatrix} \cdot \begin{pmatrix}
\frac{1}{\sqrt{8}} \\ \frac{1}{\sqrt{8}} \\ \frac{1}{\sqrt{8}} \\ \frac{1}{\sqrt{8}} \\ \frac{1}{\sqrt{8}} \\ \frac{1}{\sqrt{8}} \\ -\frac{1}{\sqrt{8}} \\ \frac{1}{\sqrt{8}} \\ \frac{1}{\sqrt{8}}
\end{pmatrix} = \begin{pmatrix}
\frac{1}{\sqrt{32}} \\ \frac{1}{\sqrt{32}} \\ \frac{1}{\sqrt{32}} \\ \frac{1}{\sqrt{32}} \\ \frac{1}{\sqrt{32}} \\ \frac{1}{\sqrt{32}} \\ \frac{5}{\sqrt{32}} \\ \frac{1}{\sqrt{32}} \\ \frac{1}{\sqrt{32}}
\end{pmatrix}$$

Die Wahrscheinlichkeit, bei der abschließenden Messung das richtige Ergebnis zu erhalten, ist $(\frac{5}{\sqrt{32}})^2$, das falsche erhält man mit Wahrscheinlichkeit $(\frac{1}{\sqrt{32}})^2$. Betrachtet man nun die Wahrscheinlichkeit,

dass der richtige Zustand x_0 mit absoluter Mehrheit gemessen wird in bezug zur Anzahl der Teilsysteme, erkennt man, dass in diesem Fall schon 15 Teilsysteme ausreichen um eine Wahrscheinlichkeit von 0,99 zu erreichen. Es ist offensichtlich, dass die 72 Teilsysteme aus-

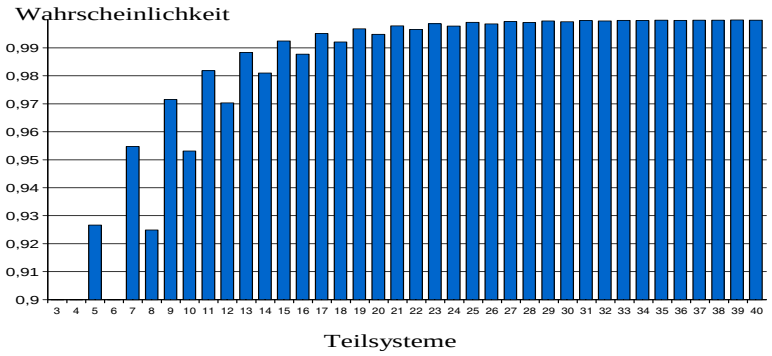

Abbildung 6: Wahrscheinlichkeitsverteilung

reichen um x_0 zu bestimmen, denn beispielsweise ist P(falscher Zustand wurde mehr als 37 mal gemessen)=10^{-3}, dagegen P(richtiger Zustand weniger als 37 mal gemessen)=10^{-10}. Damit wird das gesuchte F sicher erkannt.
Die Wahrscheinlichkeiten liegen sogar noch höher, wenn man den genaueren Ansatz der relativen Mehrheit mit bedingten Wahrscheinlichkeiten verwendet.
In Abb. 6 ist für das obige Beispiel mit $N = 8$ die Wahrscheinlichkeit, das richtige Ergebnis zu erhalten, in Abhängigkeit von der Anzahl η der Teilsysteme dargestellt. Der alternierende Verlauf in der Grafik läßt sich dadurch erklären, dass es schwieriger ist, eine absolute Mehrheit bei einer geraden Menge zu bekommen, da dort zwei Stimmen mehr benötigt. In der ungeraden Situation reicht eine Stimme aus und deshalb ist die Wahrscheinlichkeit in diesen Fällen höher.

Literatur

[Gru99] Gruska, Josef: Quantum Computing. London:McGraw-Hill, 1999

[RP98] Rieffel, Eleanor und Polak, Wolfgang: An Introduction to Quantum Computing for Non-Physicists. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9809016

[GB97] Grassl, Markus und Beth, Thomas: On the Complexity of Quantum Searching Using Complex Queries. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9706052

[Gro97] Grover, Lov K.: Quantum Computers Can Search Abritrarily Large Databases by a Single Query. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9706005

[TS98] Terhal, Barbara M. und Smolin, John A.: Single quantum querying of a database. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9705041

[SH94] Shor, Peter W.: Algorithms for Quantum Computation: Discrete Logarithm and Factoring. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9508027