QuaC Zusammenfassung

November 21, 2023

Contents

1 Grundlagen

1.1 Definitionen

Definition 1.1. $|\phi\rangle$ ist ein Spaltenvektor und $\langle\phi|$ ein Spaltenvektor.
$\langle\phi|\psi\rangle = \sum_{i=1}^{n} \phi_i^* \cdot \psi_i$ ist das innere Produkt.
$|0\rangle = \binom{1}{0}$ und $|1\rangle = \binom{0}{1}$. Das sind die Basis Vektoren.

Ein Qubit ist ein normierter Vektor der Form $\alpha|0\rangle + \beta|1\rangle$. Auf diesem kann eine Messung durchgeführt werden, sodass das Qubit kollabiert zu 0 mit Wahrscheinlichkeit $|\alpha|^2$ und mit Wahrscheinlichkeit $|\beta|^2$ zu 1.
Ein n-Qubit ist ein Vektor der Dimension 2^n. Das heißt, obiges Beispiel ist ein 1-Qubit und ein 2-Qubit hat vier Einträge. Die Vektoren sind dabei immer normiert. In höher dimensionalen Vektoren kann eine Messung einzelner Bits ausreichen, um das System zum kollabieren zu bringen.
Ein 1-Qubit-Gatter ist eine unitäre 2×2 Matrix U. Das heißt, $U^t U = I$, wobei $U^t = (U^T)^*$. Ein Spezialfall ist dabei

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

Die Transformation eines Qubits ϕ mit H ist dabei einfach $H \cdot \phi$. So ist zum Beispiel

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

Im Falle eines 2-Qubit-Gatters würde eine unitäre 4×4 Matrix herangezogen werden. Ein Beispiel ist das *C-NOT* Gatter

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Lemma 1.2. *Das innere Produkt hat folgende Eigenschaften:*

1. $\langle\phi|\phi\rangle \geq 0$
2. $\langle\phi|(a|\phi_1\rangle + b|\phi_2\rangle) = a\langle\phi|\phi_1\rangle + b\langle\phi|\phi_2\rangle$
3. $\langle\psi|\phi\rangle^* = \langle\phi|\psi\rangle$

Für eine $m \times n$ Matrix A ist weiter

$$(|A\psi\rangle, |A\phi\rangle) = \langle AA^t\psi|\phi\rangle$$

Definition 1.3. Wir definieren weiter eine Qubit-Norm $||\cdot|| : \mathbb{C}^n \to \mathbb{C}$ durch

$$|||\phi\rangle|| = \sqrt{\langle\phi|\phi\rangle}$$

Dabei heißt ein Vektor ϕ unitär, wenn $|||\phi\rangle|| = 1$.

Definition 1.4. Wir definieren zu den Vektoren $|0\rangle$ und $|1\rangle$ eine Dualbasis durch

$$|\nearrow\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$

und

$$|\searrow\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$

. Dies ist eine Orthonormalbasis.

Definition 1.5. (Tensorprodukt) Seien $x = \binom{x_1}{x_2}$ und $y = \binom{y_1}{y_2}$ zwei Vektoren. Das Tensorprodukt ist definiert als

$$x \otimes y = \begin{pmatrix}x_1y_1\\x_1y_2\\x_2y_1\\x_2y_2\end{pmatrix}$$

Das ist leicht verallgemeinerbar für höher dimensionale Vektoren. Außerdem kann das Tensorprodukt auf Matrizen

$$A = \begin{pmatrix}a_{1,1} & \dots & a_{1,n}\\ \vdots & & \vdots\\ a_{m,1} & \dots & a_{m,n}\end{pmatrix}$$

und eine beliebig dimensionale Matrix B angewandt werden durch

$$A \otimes B = \begin{pmatrix}a_{1,1}B & \dots & a_{1,n}B\\ \vdots & & \vdots\\ a_{m,1}B & \dots & a_{m,n}B\end{pmatrix}$$

Wenn mehrere Qubits $|\phi\rangle = \binom{\alpha}{\beta}$ und $|\psi\rangle = \binom{\gamma}{\delta}$ vorliegen, dann schreiben wir

$$|\phi\rangle|\psi\rangle = |\phi\psi\rangle = |\phi\rangle \otimes |\psi\rangle \in \mathbb{C}^4$$

Wird $|\phi^n\rangle$ geschrieben, so ist damit das n-fache Tensorprodukt von $|\phi\rangle$ gemeint.

Lemma 1.6. *Das Tensorprodukt hat folgende Eigenschaften:*

1. $|v\rangle \otimes |w\rangle + |v\rangle \otimes |u\rangle = |v\rangle(|w\rangle + |u\rangle)$
2. $a(|v\rangle \otimes |w\rangle = a|v\rangle \otimes |w\rangle) = |v\rangle \otimes a|w\rangle$
3. $(|u\rangle \otimes |v\rangle, |w\rangle \otimes |x\rangle) = (|u\rangle \otimes |w\rangle, |v\rangle \otimes |x\rangle)$

1.2 Quantenschaltkreise

Diese unterscheiden sich von klassischen Schaltkreisen insofern, dass Operationen reversibel sind. In klassischen Schaltkreisen ist das nicht der Fall, da zum Beispiel bei einer $\wedge$ Verknüpfung mit Ergebnis 0 nicht auf die Eingabewerte rückgeschlossen werden kann. Quantenschaltkreise können verwendet werden um sowohl klassische als auch probabilistische Schaltkreise zu modellieren.

Theorem 1.7 (no cloning satz)**.** *Es gibt keine unitäre Transformation U, das ein Qubit $|\phi\rangle$ kopiert.*

Proof. Falls so ein U existiert, dann gilt $\forall|\phi\rangle$ und $\forall|\psi\rangle$ immer $U|\phi 0\rangle = |\phi\phi\rangle$ und $U|0\psi\rangle = |\psi\psi\rangle$. Dann gilt

$$\langle\phi|\psi\rangle\langle\phi|\psi\rangle = \langle\phi\psi|\phi\phi\rangle = (U|\phi 0\rangle, U|\psi 0\rangle) = (U^t U|\phi 0\rangle, |\psi 0\rangle) = \langle\phi 0|\psi 0\rangle = \langle\phi|\psi\rangle\langle 0|0\rangle$$

Da 0 die Norm 1 hat folgt $\langle\phi|\psi\rangle = \langle\phi|\psi\rangle^2$ was nur für die Werte 0 und 1 der Fall ist. □

1.3 Simulation klassischer Schaltkreise

Wir definieren das Fredkin Gatter $f(a,b,0) = (a,b,0)$ und $f(a,b,1) = (b,a,1)$ für $a,b \in \{0,1\}$. Dafür dient die folgende 8×8 Matrix:

$$F = \begin{pmatrix} I_3 & 0 & 0 \\ 0 & J_3 & 0 \\ 0 & 0 & I_2 \end{pmatrix}$$

wo J_3 die um 90° rotierte 3×3 Einheitsmatrix ist. Mit diesem können klassische Boole'sche Funktionen simuliert werden. Mit der obigen Matrix gilt $f(0,b,c) = (c \wedge b, \overline{c} \wedge b, c)$ und ist somit eine UND-Funktion. Der erste Output ist der für die Operation relevante Teil, die anderen beiden werden für die Umkehrung gebraucht. Ein ODER-Gatter kann mit $f(1,b,c) = (b \wedge \overline{c}, b \wedge c, c)$ simuliert werden. Zum Schluss ist eine Negation durch $f(0,1,c) = (c, \overline{c}, c)$ darstellbar. Da diese drei Funktionen eine vollständige Basis sind,

kann somit jeder klassische Schaltkreis durch Verkettung von Fredkin Gattern simuliert werden. Für einen klassischen Schaltkreis c gibt es einen Quantenschaltkreis der Größe $p(|c|)$, der c berechnet.

1.4 Simulation probabilistischer Schaltkreise

Ein probabilistischer Schaltkreis ist ein klassische Schaltkreis, der als Eingabe mit gewisser Wahrscheinlichkeit eine Konstante bekommt und der das Ergebnis mit einer genügend großen Wahrscheinlichkeit berechnet.
Die Eingabe ist dabei gegeben als Input x und einer Menge z an Zufallsbits. Die Aufgabe des Schaltkreises könnte es sein, mit Wahrscheinlichkeit $p > \frac{3}{4}$ zu berechnen, ob x eine Primzahl ist.
Der Input einer Zufallsvariablen kann durch Transformation mittels eines Hadamard Gatters und anschließender Messung des Qubits simuliert werden.

1.5 Die Komplexitätsklasse BQP

Im klassischen Sinne ist ein Problem L einer Sprache Σ^* in der Klasse P, wenn es einen polynomiellen Algorithmus gibt, der L entscheidet.

Definition 1.8 (BPP - bounded error probabilistic polynomial time)**.** Sei $L \subseteq \Sigma^*$. Dann gilt $L \in$BPP genau dann, wenn es eine probabilistische Turing Maschine mit Zufallsbits und ein Polynom p gibt, sodass $\forall x \in \Sigma^*$

- $\forall x$ gilt $x \in L$ impliziert $p[c_n(x) = 1] \geq \frac{3}{4}$
- $\forall x$ gilt $x \notin$ ipliziert $p[c_n(x) = 0] \geq \frac{3}{4}$
- c_n hat höchstes $c(n)$ Gatter

Definition 1.9 (BQP - bounded error quantum polynomial time)**.** Sei $L \subseteq \Sigma^*$. Dann gilt $L \in$BQP genau dann, wenn es eine Familie von Quantenschaltkreisen $\{c_1, ...\}$ und ein Polynom p gibt, sodass $\forall x \in \Sigma^*$

- $\forall x$ gilt $x \in L$ impliziert $p[c_n(x) = 1] \geq \frac{3}{4}$
- $\forall x$ gilt $x \notin$ impliziert $p[c_n(x) = 0] \geq \frac{3}{4}$
- c_n hat höchstes $c(n)$ Gatter

Remark 1.10. Es gilt BQP$\subset$BPP.

Definition 1.11. Sei S eine Menge an Transformationen. S ist eine universelle Menge für alle U unitär, wenn U mit Gattern aus S approximiert werden kann. D.h.

$$\forall \epsilon > 0 \; \exists G_1, G_2, ...G_k \in S \text{ s.d. } ||U - G_1 G_2 ... G_k|| < \epsilon$$

Ein Beispiel für S ist CNOT mit allen 1-Qubit Gattern. Diese ist aber unendlich groß. $S = \{\text{CNOT, H, } \frac{\pi}{8} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{pi}{4}} \end{pmatrix}\}$ ist ein endliches Beispiel.

Definition 1.12 (PP)**.** $L \in PP$ gilt genau dann, wenn es eine polynomiell beschränkte nicht-deterministische Turin Maschine M gibt sodass,

- $x \in L \Rightarrow$ die Anzahl der akzeptierenden Pfade in $M(x)$ ist größer als die Anzahl der verwerfenden Pfade in $M(x)$
- $x \notin L \Rightarrow$ die Anzahl der akzeptierenden Pfade in $M(x)$ ist höchstens die Anzahl der verwerfenden Pfade in $M(x)$

Das ist äquivalent zu $L \in PP \Leftrightarrow \exists f \in \#P, g \in FP$ sodass $\forall x \in L \Leftrightarrow f(x) \geq g(x)$.

Wir wollen nun umgekehrt zeigen, dass jeder Quantenschaltkreis durch klassische Schaltkreise simuliert werden kann. Dafür betrachten wir $U = \{H, R, CNOT, Toffoli\}$, eine universelle Menge von Quantengatter. Mit Hilfe dieser Gatter ist es möglich, zu beweisen, dass $BQP \subseteq PP$.

2 Erste Quantenalgorithmen

Im Folgenden ist eine Relation für die n-te Hadamard Matrix H_n wichtig. Man nummeriere die Zeilen dieser Matrix mit binär Zahlen von 0000... bis 111... als x und die Spalten auf die selbe Weise als y. Es gilt dann $(H_n)_{x,y} = \frac{1}{\sqrt{2^n}}(-1)^{x \cdot y}$. Dabei wird $x \cdot y$ definiert als

$$x \cdot y = \bigoplus_{i=1}^{n} x_i \wedge y_i$$

Wenn angenommen wird, dass für eine zu berechnende Funktion f eine black box mit einem Quantenschaltkreis U_f existiert, so soll die Anzahl der benötigten Anfragen an den Schaltkreis U_f bestimmt werden.

2.1 Algorithmus von Deutsch

Sei $f : \{0,1\} \to \{0,1\}$. Wir wollen $f(0) \oplus f(1)$ berechnen. Im klassischen Fall müssen zwei Anfragen an f gestellt werden, um $f(0)$ und $f(1)$ zu bestimmen. Der Deutsch

f(0)	f(1)
0	0
0	1
1	0
1	1

Algorithmus löst das mit einer Abfrage wie folgt:
Die Eingaben $|0\rangle$ und $|1\rangle$ werden Hadamar-transformiert, d.h.

$$|01\rangle \overset{H}{\to} \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = \frac{1}{2}(|00\rangle + |10\rangle - |01\rangle - |11\rangle)$$

Nun gibt es vier verschiedene Fälle für f wie in der obigen Tabelle. Allgemein folgt aber $\overset{U_f}{\to} \frac{1}{2}(|0(0 \oplus f(0))\rangle + |1(0 \oplus f(1))\rangle - |0(1 \oplus f(0))\rangle - |1(1 \oplus f(1))\rangle)$

2.2 Deutsch-Josza-Problem

Sei $f : \{0,1\}^n \to \{0,1\}$. f ist von Typ 1, wenn $f(x) = 1 \vee f(x) = 0 \; \forall x$. f ist von Typ 2, wenn $|\{x : \; f(x) = 0\}| = |\{x : f(x) = 1\}| = 2^{n-1}$, sich die Funktion also im Gleichgewicht befindet. Im klassischen Fall sind bis zu $2^{n-1} + 1$ Abfragen nötig, um den Typen on f zu bestimmen.
In einem probabilistischen Schaltkreis werden zufällig $x \in \{0,1\}^n$ gewählt und berechne $f(x)$. Falls alle $f(x)$ gleich sind, ist mit hoher Wahrscheinlichkeit eine Funktion von Typ 1 vorliegend. Werden nun k Eingaben zufällig und "ohne Zurücklegen" gezogen, dann ist die Fehlerwahrscheinlichkeit, wenn k mal der selbe Funktionswert zurückgegeben wird, gegeben durch

$$\prod_{i=1}^{k} \frac{2^{n-1} - i}{2^n - i} < \frac{1}{2^k}$$

Mit dieser Ungleichung ist es leicht, ein k zu finden, wodurch die Fehlerwahrscheinlichkeit kleiner als ein gegebenes ε ist.
Der Quantenschaltkreis nimmt als Input $|0^n\rangle$ und $|1\rangle$. Beide werden Hadamard-transformiert, wobei $H|0^n\rangle$ bereits in einer früheren Vorlesung berechnet wurde. Es ergibt sich

$$|0^n\rangle|1\rangle \overset{H}{\to} \frac{1}{\sqrt{(2^n)}} \sum_{x \in \{0,1\}^n} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

Mit der Anwendung des Schaltkreises geht dies weiter zu

$$\overset{U_f}{\to} \frac{1}{\sqrt{(2^n)}} \sum_{x \in \{0,1\}^n} |x\rangle \left(\frac{|f(x)\rangle - \overline{|f(x)\rangle}}{\sqrt{2}} \right)$$

Weitere Vereinfachungen liefern anschließend

$$\frac{1}{2^n} \sum_{z \in \{0,1\}} \alpha_z |z\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

wobei

$$\alpha_z = \sum_{x \in \{0,1\}^n} (-1)^{xz \oplus f(x)}$$

Mit nur einer Abfrage kann der Typ bestimmt werden, wodurch die Funktion bereits eindeutig definiert ist. Mit diesem Quantenparallelismus ist also nur eine Messung nötig.

2.3 Simon's Algorithmus

Wir betrachten $f : \{0,1\}^n \to \{0,1\}^n$ und $a \in \{0,1\}^n$ und wir repräsentieren mit $\oplus$ das bitweise XOR. f besitzt diese Eigenschaften:

1. $\forall x \in \{0,1\}^n,\ f(x) = f(x \oplus a)$
2. $\forall x, y$ mit $y \neq x$ und $y \neq x \oplus a$ gilt $f(x) \neq f(y)$. D.h. es liegt eine 2-zu-1 Funktion vor

Das Problem ist, f zu finden.

Remark 2.1 (Klassischer Ansatz). Für jedes x lässt sich $f(x)$ berechnen. D.h. finden wir x_1 und x_2 mit $f(x_1)0f(x_2)$, so erhalten wird $x_1 = x_2 \oplus a$ und damit $a = x_1 \oplus x_2$.
Generell gilt, werden k verschiedene Inputs probiert und ist $f(x_i) \neq f(x_j)$ für alle $i, j \leq k$, so können $\binom{k}{2}$ Werte ausgeschlossen werden. Damit nur ein a übrig bleibt, müssen also so viele k probiert werden, damit

$$2^n - 1 - \binom{k}{2} = 1$$

gilt. D.h. $k = \Omega(\sqrt{2^n})$.

Remark 2.2 (Probabilistischer Ansatz). Sei a' zufällig in $\{0,1\}^n \setminus 0^n$. Es gilt $\mathbb{P}[a' = a] = \frac{1}{2^n - 1}$ ist unabhängig von a'. Wir nehmen wieder k Stichproben mit paarweise verschiedenen Funktionswerten $x_1, ..., x_k$, sodass gilt $x_i \neq x_j$ für alle $i, j \leq k$ an. Nun gilt $\mathbb{P}[\underbrace{a = a'}_{=A} \mid \underbrace{f(x_i) \neq f(x_j),\ x_i \neq x_j}_{=B}]$. Nun ist bekannt, dass $P[B|A] = 1$. $\mathbb{P}[B]$ ist unabhängig von a'. Es folgt daher, dass $P[A|B]$ komplett unabhängig von der Wahl von a' ist. Eben da diese Wahrscheinlichkeit für alle noch nicht getesteten a' gleich ist, müssen weiterhin $\sqrt{2^n}$ Stichproben gemacht werden.

Remark 2.3 (Quanten Ansatz). Der Schaltkreis besteht aus zwei mal der Eingabe $|0^n\rangle$, wobei der Ablauf so aussieht

$$|0^n\rangle|0^n\rangle \overset{H_n}{\to} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0^n\rangle \overset{U_f}{\to} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|f(x)\rangle \overset{M_n}{\to} \frac{|x\rangle|f(x)\rangle + |x\oplus a\rangle|f(x)\rangle}{\sqrt{2}}$$

Durch eine weitere Hadamard Transformation auf dem ersten n-Qubits erhält man daraus

$$\frac{1}{\sqrt{2^n}} \left(\sum_{z\in\{0,1\}^n} (-1)^{zx}(1+(-1)^{za})|z\rangle \right) |f(x)\rangle = \sum_{z\in\{0,1\}^n} \alpha_z |z\rangle|f(x)\rangle$$

mit

$$\alpha_z = \frac{(-1)^{zx}(1+(-1)^{za})}{\sqrt{2^{n+1}}}$$

Nun gilt, falls $za = 1$, so gilt $\alpha_z = 0$. Darum messen wir nur z mit $za = 0$, gilt $\alpha_z = \frac{(-1)^{zx}}{\sqrt{2^{n-1}}}$. D.h. $|\alpha_z|^2$ ist gleich für alle z mit $za = 0$.
Bemerke nun, dass $za = 0 \Rightarrow \bigoplus_{i=1}^n z_i \wedge a_i = 0$. Das heißt, z ist orthogonal zu a. Um genau zu sein, erhält man diese Relation für alle $n-1$ Vektoren z^k, die orthogonal zu a sind. Formaler ist $\{z \in \{0,1\}^n : \ za = 0\}$ ist ein Untervektorraum mit Dimension $n-1$. Wir brauchen also $n-1$ l.u. z mit $za = 0$, um das System lösen zu können und a zu berechnen. Dafür sind die folgenden Schritte da.
Schritt 1 Generiere $z_1 \neq 0^n$ mit $z_1 a = 0$. Das hat Wahrscheinlichkeit $1 - \frac{1}{2^{n-1}}$.
Schritt 2 Generiere z_2 mit $z_2 a = 0$ das linear unabhängig zu z_1 und 0^n ist. Das hat Wahrscheinlichkeit $1 - \frac{2}{2^{n-1}}$
Schritt k Generiere z_k mit $z_k a = 0$ das linear unabhängig zu dem Raum erzeugt von den vorherigen $k-1$ Vektoren und 0^n ist. Das hat Wahrscheinlichkeit $1 - \frac{2^{k-1}}{2^{n-1}}$.
Wir sehen, dass die Wahrscheinlichkeit $n-1$ solche Vektoren zu generieren

$$\prod_{j=1}^{n-1} (1 - \frac{2^{j-1}}{2^{n-1}}) \leq \frac{1}{4}$$

ist. Wird dieses Experiment k mal wiederholt, so ist die Wahrscheinlichkeit, dass alle Experimente Vektoren produzieren, die linear abhängig sind, $\frac{3}{4}^k \leq \frac{1}{2^{k-2}}$. Bei $k(n-1)$ Stichproben, findet man a mit Wahrscheinlichkeit $\geq 1 - \frac{1}{2^{k-2}}^k$.

3 Suchalgorithmus von Grover

Sei $f : \{0,1\}^n \to \{0,1\}$ und U_f ein Schaltkreis, der f berechnet. Gesucht ist ein $a \in \{0,1\}^n$ mit $f(a) = 1$. Es kann angenommen werden, dass es nur genau ein a mit

$f(a) = 1$ gibt. Wir definieren für leichtere Schreibweise $2^n = N$. Wir interessieren uns für die Anzahl an Anfragen an f. Im klassischen Fall müssen damit bis zu $N-1$ Anfragen gestellt werden. Im probabilistischen Fall ist der Erwartungswert $\frac{2N-1}{2}$. Im Quantenfall genügen hingegen $\mathcal{O}(\sqrt{N})$ Anfragen. Im Folgenden wird das bewiesen.

Definition 3.1 (D Transformation)**.**

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

Diese Matrix wird Diffusions-Matrix genannt. Man sieht, dass für

$$D \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_N \end{pmatrix}$$

Wobei $\beta_i = \frac{2}{N}\sum_{j=1}^{N} \alpha_k - \alpha_i$. Das heißt für den Durchschnitt der Amplituden $\mu = \sum_{j=1}^{N} \frac{\alpha_j}{N}$ ist $\beta_i = 2\mu - \alpha_i$.

Als Eingabe Qubits dienen wieder $|0^n\rangle$ und $|0\rangle$. Diese werden Hadamard transformiert und das Resultat in U_f eingegeben. Der obere Output von U_f wird D transformiert. U_f mit anschließender D Transformation wird G genannt. Der Grover Algorithmus funktioniert durch m-males Anwenden von G. Zusammengefasst:

$$|0^n\rangle|0\rangle \xrightarrow{H} \frac{1}{\sqrt{N}} \sum_{z\in\{0,1\}^n} |z\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{N}} \sum_{z\in\{0,1\}^n} (-1)^{f(z)}|z\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)$$

Durch t-maliges Anwenden von G sind die Werte α_x^t gegeben. Insbesondere ist $\alpha_a^1 = \frac{3}{\sqrt{N}} - \frac{4}{N\sqrt{N}}$ und $\alpha_x^1 = \frac{1}{\sqrt{N}} - \frac{4}{N\sqrt{N}}$ für $x \neq a$. Generell ist

$$\alpha_a^t = 2\mu_{t-1} + \alpha_a^{(t+1)} = -\frac{2}{N}\alpha_a^{(t-1)} + \left(1 - \frac{2}{N}\right)\alpha_x^{(t-1)}$$

wobei μ_t der Durchschnitt der Amplituden nach t maliger Anwendung von G sei. Lösen liefert $\alpha_a^t = \sin((2t+1)\Theta)$ und $\alpha_x^t = \frac{1}{\sqrt{N-1}}\cos((2t+1)\Theta)$ wobei $sin^2(\Theta) = \frac{1}{N}$. Nun soll die Wahrscheinlichkeit für α_x^t minimiert werden, d.h. gesucht ist ein Θ, sodass $\cos((2t+1)\Theta) = 0$. Für $t = \frac{\pi}{4\Theta} - \frac{1}{2} + m\frac{\pi}{\Theta}$ ist der Cosinus gleich 0. Das heißt für großes N genügt $t = \lfloor \frac{\pi}{4\Theta} \rfloor$ damit der Cosinus ungefähr 0 ist. t ist damit in $\mathcal{O}(\sqrt{N})$.

Im allgemeinen Fall, wenn $|\{x|f(x)=1\}|=M$, dann findet man mit großer Wahrscheinlichkeit ein x mit $f(x)=1$ in $t=\sqrt{\frac{N}{M}}$ Iterationen. Geometrisch kann man sich den Algorithmus als eine Rotation der Qubits vorstellen.

3.1 Untere Schranke für den Grover-Suchalgorithmus

Für die untere Schranke wird ein Algorithmus A betrachtet, der bildlich gesprochen Fragen an U_f stellen kann. Dafür steht im Folgenden O, also Orakel.
Die Behauptung ist, dass A $\Omega(\sqrt{N})$ Anfragen an U_f stellen muss, um a zu bestimmen. A kann wie folgt geschrieben werden:

$$\underbrace{U_T O U_{T-1} ... U_1 O U_1 O}_{A} |0^n\rangle$$

Die Anzahl an Fragen ist daher T. Für den Beweis wird das folgende Lemma benötigt:

Lemma 3.2. *Seien* $|\phi\rangle=\sum_{i=1}^n \alpha_i|i\rangle$ *und* $|\psi\rangle=\sum_{i=1}^n \beta_i|i\rangle$ *zwei Quantenzustände, sodass*

$$\sum_{i=1}^{n} \left| |a_i|^2 - |\beta_i|^2 \right| > \varepsilon$$

dann ist

$$|||\phi\rangle - |\psi\rangle|| > \frac{\varepsilon}{2}$$

Wir definieren die Funktion $f(z)=0\ \forall z$ und das dazugehörige Orakel O_f. Wir definieren $|\psi_t\rangle = U_t O U_{t-1} ... U_1 O U_1 O_A |0^n\rangle = \sum_{i=1}^n \alpha_i^t |i\rangle$. Außerdem definieren wir $q_x(\psi_t) = |\alpha_x^t|^2$. Man kann sehen, dass

$$\sum_{t=1}^{T} \sum_{x\in\{0,1\}^n} q_x(\psi_t) = T = \sum_{x\in\{0,1\}^n} \sum_{t=1}^{T} q_x(\psi_t) \Rightarrow\ \exists x_0 :\ \sum_{x\in\{0,1\}^n} \sum_{t=1}^{T} q_{x_0}(\psi_t) \leq \frac{T}{N}$$

Damit wird eine neue Funktion definiert

$$g(x) = \begin{cases} 1,\ x = x_0 \\ 0,\ x \neq x_0 \end{cases}$$

Dann sei $|\psi'_T\rangle = U_T O_g U_{t-1} O_g ... U_1 O_g U_0 |0^n\rangle$. Nach einer Messung erkennt man bereits einen Unterschied in den beiden Schaltkreisen, woraus folgt, dass

$$\sum_{i=1}^{n} \left| |\alpha_i|^2 - |\alpha'_i|^2 \right| > \varepsilon > 0$$

Wir schließen mit obigem Lemma

$$|||\psi_T\rangle - |\psi'_T\rangle|| > \frac{\varepsilon}{2}$$

Wir definieren aus diesen beiden Zuständen einen neuen Zustand

$$|\psi_T\rangle_i = U_T O_g U_{T-1} O_g ... O_g U_i O_f U_{i-1} O_f ... O_f U_1 O_f U_0 |0^n\rangle$$

Wir finden

$$\varepsilon < |||\psi_T\rangle - |\psi'_T\rangle|| < |||\psi_T\rangle_T - |\psi_T\rangle_0|| = |||\psi_T\rangle_T - |\psi_T\rangle_i + |\psi_T\rangle_i - |\psi_T\rangle_0|| = ||\sum_{t=1}^{T} |\psi_T\rangle_t - |\psi_T\rangle_{t-1}||$$
$$\overset{\Delta}{\leq} \sum_{t=1}^{T} ||\underbrace{|\psi_T\rangle_t - |\psi_T\rangle_{t-1}}_{=E_i}|| = \sum_{t=1}^{T} ||E_i||$$

Zur Vereinfachung schreiben wir nun:
$|\psi_T\rangle_i = \underbrace{U_T O_g ... O_g U_i}_{=U} O_f \underbrace{U_{i-1} ... O_f U_0}_{=|\psi_i\rangle} |0^n\rangle$ und
$|\psi_T\rangle_{i-1} = \underbrace{U_T O_g ... O_g U_i}_{=U} O_g U_{i-1} ... O_f U_0 |0^n\rangle$. Es gilt dann $|\psi_T\rangle_{i-1} = U O_g |\psi_i\rangle = U(|\psi_i\rangle - 2\alpha^i_{x_0}|x_0\rangle)$, also

$$\sum_{t=1}^{T} ||E_i|| \leq \sqrt{\sum_{t=1}^{T} ||E_i||^2 T} = \sqrt{\sum_{t=1}^{T} 4|\alpha^i_{x_0}|^2 T} = 2\sqrt{\sum_{i=1}^{n} q_{x_0}(\psi_i) T} \leq 2\frac{T}{\sqrt{N}}$$

Geht man zurück zu oberer Forderung mit ε, so findet man durch gegenüberstellen der beiden Ungleichungen $T = \frac{\varepsilon\sqrt{N}}{4} \in \Omega(\sqrt{N})$.

3.2 Anwendungen vom Grover-Algorithmus

Definition 3.3 (QSEARCH)**.** Sei $f : \{0,1\}^n \to \{0,1\}^n$ und sei $M = |\{x \in \{0,1\}^n | f(x) = 1\}|$. Wenn $M \neq 0$, dann nehmen wir an, dass QSEARCH ein x mit $f(x) = 1$ in erwarteter Zeit $\mathcal{O}(\sqrt{\frac{N}{M}})$ findet. Alle solche x sind gleich wahrscheinlich. Wenn $M = 0$, dann hält QSEARCH nicht.

3.2.1 Minimumsproblem

Sei $f : \{0, ..., N-1\} \to \{0, ..., M\}$ injektiv. Finde x, sodass $\forall y \neq x$ gilt $f(x) < f(y)$. Als Komplexitätsmaß wird die Anzahl der Vergleiche herangezogen. Wir werden sehen, dass der Quanten-Ansatz $\Theta(\sqrt{N})$ Vergleiche benötigt.

Definiere für ein festes y die Funktion

$$g_y(x) = \Big\{1, f(x) < f(y)0, \text{ sonst}$$

Wir definieren den Schaltkreis U_g für g mit den Inputs x, y und einem zusätzlichen Qubit b und den Ausgängen x, y und $b \oplus g_y(x)$. Der Algorithmus funktioniert schließlich wie folgt:

1. Wähle $y \in_R \{0, .., N-1\}$
2. REPEAT
3. $|\Phi\rangle := (H_n|0^n\rangle)|y\rangle|1\rangle$
4. $QSEARCH(g_y(x))$
5. $y' =$ MEssung der ersten n Register
6. IF $f(y') < f(y)$ THEN $y = y'$
7. t=t+1
8. UNTIL $t > l$
9. AUSGABE $(y, f(y))$

Dabei ist l ein Parameter, der später fixiert wird.

Lemma 3.4. *Sei $p(q, r)$ die Wahrscheinlichkeit, dass der Algorithmus das r-kleinste Element bei q Elementen wählt. Dann ist $p(q, r) = \frac{1}{r}$, wenn $r \leq q$ und 0 sonst.*

Lemma 3.5. *Die erwartete Anzahl an Fragen an U_g bis das Argminimum y gefunden wird, ist $\mathcal{O}(\sqrt{N})$.*

3.2.2 Untere Schranke

Sei $g : \{0, ..., N-1\} \to \{0, ..., 2N-1\}$ und $g(i) = i+N(1-f(i))$ wobei $f : \{0, ..., N-1\} \to \{0, 1\}$. Falls $\forall x,\ f(x) = 0 \Leftrightarrow g(i) = i + N$. Falls $\exists x,\ f(x) = 1$, dann $g(x) = x$. Es gilt

$$\min g(x) \geq N \Leftrightarrow\ \forall x\ f(x) = 0$$

und

$$\min g(x) < N \Leftrightarrow \exists x :\ f(x) = 1$$

Würde nun ein Algorithmus existieren, der weniger als $\Omega(\sqrt{N})$ Laufzeit benötigt, dann könnte Grover's Algorithmus genauso verbessert werden, was nicht möglich ist.