

QuaC Zusammenfassung

February 8, 2024

Contents

1	Grundlagen	3
1.1	Definitionen	3
1.2	Quantenschaltkreise	5
1.3	Simulation klassischer Schaltkreise	5
1.4	Simulation probabilistischer Schaltkreise	6
1.5	Die Komplexitätsklasse BQP	6
2	Erste Quantenalgorithmen	7
2.1	Algorithmus von Deutsch	7
2.2	Deutsch-Josza-Problem	8
2.3	Simon's Algorithmus	9
3	Suchalgorithmus von Grover	10
3.1	Untere Schranke für den Grover-Suchalgorithmus	12
3.2	Anwendungen vom Grover-Algorithmus	13
3.2.1	Minimumsproblem	13
3.2.2	Untere Schranke	14
3.2.3	Element Distinctness	15
4	Shor Algorithmus	15
4.1	Diskrete Fourier Transformation	15
4.2	Quantum Fourier Transformation	15
4.3	Algorithmus für die Faktorisierung	16
4.4	Algorithmus für das Ordnungsproblem	17
5	Problem der verborgenen Untergruppe	21
5.1	Graphenautomorphie Problem	23
6	Endliche Quantenautomaten	23
6.1	Messung nach Lesen von jedem Symbol	25
6.2	Beschreibung der Berechnung	25
6.3	Der Two-Way-Quanten-Automat	26
7	Quantenkommunikationskomplexität	27
7.1	klassische Varianten	27
7.2	probabilistische Varianten	27

7.3	Quantenkommunikationskomplexität	28
7.4	Finger Printing	32
7.5	Quantenkryptologie	34
8	Quanten fehlerkorrigierende Codes	35
8.1	klassische fehlerkorrigierende Codes	35
8.2	Quantencodes	35
8.2.1	Bit-Flips	36
8.2.2	Phase-Flip	36
8.2.3	9-Bit Code	36
8.2.4	CSS Codes	36

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.

$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. 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 höherdimensionalen 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 $|| \cdot || : \mathbb{C}^n \rightarrow \mathbb{C}$ 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 $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ und $y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ zwei Vektoren. Das Tensorprodukt ist definiert als

$$x \otimes y = \begin{pmatrix} x_1 y_1 \\ x_1 y_2 \\ x_2 y_1 \\ x_2 y_2 \end{pmatrix}$$

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 $|\phi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ und $|\psi\rangle = \begin{pmatrix} \gamma \\ \delta \end{pmatrix}$ 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. \square

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 $f(0, b, c) = (c \wedge b, \bar{c} \wedge b, c)$ 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 $f(1, b, c) = (b \wedge \bar{c}, b \wedge c, c)$ simuliert werden. Zum Schluss ist eine Negation durch $f(0, 1, c) = (c, \bar{c}, c)$ 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 klassischer 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 $L \in \text{BPP}$ 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 $x \notin L$ impliziert $p[c_n(x) = 0] \geq \frac{3}{4}$
- c_n hat höchstes $c(n)$ Gatter

Definition 1.9 (BQP - bounded error quantum polynomial time). Sei $L \subseteq \Sigma^*$. Dann gilt $L \in \text{BQP}$ genau dann, wenn es eine Familie von Quantenschaltkreisen $\{c_1, \dots\}$ 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 $x \notin L$ impliziert $p[c_n(x) = 0] \geq \frac{3}{4}$
- c_n hat höchstes $c(n)$ Gatter

Remark 1.10. Es gilt $\text{BQP} \subseteq \text{BPP}$.

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.

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

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 $f : \{0, 1\} \rightarrow \{0, 1\}$. 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.

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

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

2.2 Deutsch-Josza-Problem

Sei $f : \{0, 1\}^n \rightarrow \{0, 1\}$. 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 von 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

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

Mit der Anwendung des Schaltkreises geht dies weiter zu

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

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 $f : \{0,1\}^n \rightarrow \{0,1\}^n$ 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 $f(x_1) = f(x_2)$, so erhalten wir $x_1 = x_2 \oplus a$ 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, \dots, x_k , sodass gilt $x_i \neq x_j$ für alle $i, j \leq k$ an. Nun gilt $\mathbb{P}[\underbrace{a = a'}_{=A} | \underbrace{f(x_i) \neq f(x_j), x_i \neq x_j}_{=B}]$. 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

$$|0^n\rangle|0^n\rangle \xrightarrow{H_q} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|0^n\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle \xrightarrow{M_q} \frac{|x\rangle|f(x)\rangle + |x \oplus a\rangle|f(x)\rangle}{\sqrt{2}}$$

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

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

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 $f : \{0,1\}^n \rightarrow \{0,1\}$ 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).

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

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 $\beta_i = \frac{2}{N} \sum_{j=1}^N \alpha_j - \alpha_i$. 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 -maliges 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:

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

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

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

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 $|\psi_t\rangle = U_t O U_{t-1} \dots U_1 O U_1 O_A |0^n\rangle = \sum_{i=1}^n \alpha_i^t |i\rangle$. 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 $|\psi'_T\rangle = U_T O_g U_{T-1} O_g \dots U_1 O_g U_1 |0^n\rangle$. Nach einer Messung erkennt man bereits einen Unterschied in den beiden Schaltkreisen, woraus folgt, dass

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

Wir schließen mit obigem Lemma

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

Wir definieren aus diesen beiden Zuständen einen neuen Zustand

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

Wir finden

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

Zur Vereinfachung schreiben wir nun:

$$|\psi_T\rangle_i = \underbrace{U_T O_g \dots O_g U_i}_{=U} O_f \underbrace{U_{i-1} \dots O_f U_0}_{=|\psi_i\rangle} |0^n\rangle \text{ und}$$

$$|\psi_T\rangle_{i-1} = \underbrace{U_T O_g \dots O_g U_i}_{=U} O_g U_{i-1} \dots O_f U_0 |0^n\rangle. \text{ Es gilt dann } |\psi_T\rangle_{i-1} = U O_g |\psi_i\rangle = U(|\psi_i\rangle -$$

$2\alpha_{x_0}^i |x_0\rangle$), also

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

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 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $f : \{0, \dots, N-1\} \rightarrow \{0, \dots, M\}$ 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

$$g_y(x) = \begin{cases} 1, & f(x) < f(y) \\ 0, & \text{sonst} \end{cases}$$

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 $y \in_R \{0, \dots, N-1\}$
2. REPEAT
3. $|\Phi\rangle := (H_n|0^n\rangle)|y\rangle|1\rangle$
4. $QSEARCH(g_y(x))$
5. $y' = \text{Messung der ersten } n \text{ Register}$
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 $g : \{0, \dots, N-1\} \rightarrow \{0, \dots, 2N-1\}$ und $g(i) = i + N(1 - f(i))$ wobei $f : \{0, \dots, N-1\} \rightarrow \{0, 1\}$. 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.

3.2.3 Element Distinctness

Sei $f : \{0, \dots, N-1\} \rightarrow \{0, \dots, M-1\}$ mit $M > N$. Das Problem ist, zu bestimmen, ob f injektiv ist.

4 Shor Algorithmus

4.1 Diskrete Fourier Transformation

Ein Vektor (x_0, \dots, x_{N-1}) wird durch DFT transformiert auf (y_0, \dots, y_{N-1}) wobei

$$y_j = \sum_{k=0}^{N-1} \exp\left(\frac{2\pi i j k}{N}\right) x_k = \sum_{k=0}^{N-1} x_k (w_{k,N})^j$$

wobei definiert wird

$$\exp\left(\frac{2\pi i k}{N}\right) := w_{k,N}$$

Mit $\exp(\theta i) = \cos(\theta) + i \sin(\theta)$ kann die Darstellung weiter vereinfacht werden. Die Transformation kann auf eine Matrixmultiplikation zurückgeführt werden. Mit der Fast Fourier Transformation (FFT) kann diese in $\mathcal{O}(N \log N)$ Zeit ausgeführt werden.

4.2 Quantum Fourier Transformation

Diese ist definiert durch

$$|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} (w_{k,N})^j |k\rangle$$

Insbesondere

$$\sum_{j=0}^{N-1} x_j |j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \sum_{k=0}^{N-1} (w_{k,N})^j |j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} x_j (w_{k,N})^j |x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k |k\rangle$$

Diese kann man wieder als Matrix $F_{N \times N}$ schreiben, wobei

$$(F_{N \times N})_{a,b} = \frac{1}{\sqrt{N}} (w_{1,N})^{ab}$$

Diese kann in kleinere Matrizen zerlegt werden. Dafür schreiben wir

$$|j\rangle = |j_1, \dots, j_N\rangle \quad j_i \in \{0, 1\}$$

Die Darstellung als Quantenschaltkreis ist kompliziert. Wir schreiben

$$b(0.j_l \dots j_N) = \sum_k = l^N j_k \frac{1}{2^{k-l+1}}$$

und bemerken $b(0.j_1 \dots j_N) = j$. Die QFT kann nun geschrieben werden als

$$QFT(|j\rangle) = \frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi i b(0.j_N)}|1\rangle)(|0\rangle + e^{2\pi i b(0.j_{N-1}j_N)}|1\rangle) \dots (|0\rangle + e^{2\pi i b(0.j_1 \dots j_N)}|1\rangle)$$

4.3 Algorithmus für die Faktorisierung

Sei $N \in \mathbb{N}$ teilbar. Gesucht sind $x \in \mathbb{N} \setminus \{1, N\}$ sodass $x|N$. Die besten klassischen Algorithmen brauchen $\mathcal{O}(2^{|N|})$, wobei $|N|$ die Anzahl der Bits von N ist, d.h. $|N| \approx \log N$. Dieses Problem ist Grundlage für viele andere Probleme, so zu, Beispiel das Ordnungsproblem:

Definition 4.1 (Ordnungsproblem). Gegeben $Y, N \in \mathbb{N}$, $Y < N$ und $ggT(Y, N) = 1$. Finde

$$\arg \min_r \{Y^r \equiv 1 \pmod{N}\}$$

So ein r muss existiert immer, denn Y und N sind teilerfremd.

Lemma 4.2. Sei $N \in \mathbb{N}$ teilbar. Falls $\exists x$ sodass $x^2 \equiv 1 \pmod{N}$ und $x \not\equiv 1 \pmod{N}$ und $x \not\equiv -1 \pmod{N}$, dann $ggT(x, N) \neq 1$ und $ggT(x, N) \neq N$.

Lemma 4.3. Sei $N = p_1^{e_1} p_2^{e_2} \dots p_m^{e_m}$ die Primfaktorzerlegung von N . Im Allgemeinen können wir annehmen $N \neq 2$. Definiere r_Y als die Ordnung $\text{ord}(Y, N)$ eines Elementes Y . Unter Gleichverteilung wähle der Elemente Y mit $ggT(Y, N) = 1$ gilt

$$\mathbb{P}[r_Y \text{ gerade und } Y^{\frac{r_Y}{2}} \not\equiv \pm 1 \pmod{N}] \geq 1 - 2^{-m} \geq \frac{1}{2}$$

Auf dieser Basis kann ein Algorithmus für die Faktorisierung gefunden werden:

1. Eingabe N
2. Ausgabe Faktor von N oder ? falls keiner existiert
3. IF N gerade Then Ausgabe 2
4. ELSE WÄHLE $y \in_R [3, \dots, N-1]$
5. IF $ggT(N, Y) > 1$ Then Ausgabe

6. ELSE $r_Y = \text{ord}(Y, N)$
7. IF r_Y gerade und $y^{\frac{r_Y}{2}} \not\equiv \pm 1 \pmod{N}$ Then Ausgabe $ggT(N, y^{\frac{r_Y}{2}} + 1)$
8. ELSE Ausgabe ?

Dieser Algorithmus findet einen Faktor mit Wahrscheinlichkeit $\geq \frac{1}{2}$, durch k -fache Wiederholung kann die Fehlerwahrscheinlichkeit klein gemacht werden.

4.4 Algorithmus für das Ordnungsproblem

Gegeben Y, N mit $ggT(Y, N) = 1$. Es sei $Q \gg N$ (z.B. $Q = N^2$), $n = |N|$ und $q = |Q|$. Als Eingabe fungieren zwei Register $|0^q\rangle$ und $|0^n\rangle$. Nun wird

1. Fourier Transformation auf das erste Register angewandt, dann erhält man

$$\frac{1}{\sqrt{Q}} \sum_{k=0}^{Q-1} |k\rangle |0^n\rangle$$

Das ist eigentlich eine Hadamard Transformation

2. die modulare Potenzfunktion $f_{Y,N}(k) = Y^k \pmod{N}$ auf Register 2 berechnet

$$\frac{1}{\sqrt{Q}} \sum_{k=0}^{Q-1} |k\rangle |Y^k \pmod{N}\rangle$$

3. eine Messung auf dem zweiten Register durchgeführt.

$$\frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |jr + k_0\rangle |Y^{k_0} \pmod{N}\rangle$$

für $A = \frac{Q}{r}$. Wenn r die Ordnung ist, dann gibt es nämlich r unterschiedliche Werte für $Y^k \pmod{N}$.

4. QFT auf dem ersten Register angewandt.

$$\frac{1}{\sqrt{Q}} \sum_{l=0}^{Q-1} \left(\frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} e^{\frac{2\pi i(jr+k_0)l}{Q}} |l\rangle |Y^{k_0} \pmod{N}\rangle \right)$$

5. eine Messung auf dem ersten Register ausgeführt. Dadurch erhält man ein l mit

Wahrscheinlichkeit

$$\left| \frac{1}{\sqrt{QA}} \sum_{j=0}^{A-1} e^{\frac{2\pi i(jr+k_0)l}{Q}} \right|^2 = \frac{1}{QA} \left| e^{\frac{2\pi i k_0 l}{Q}} \sum_{j=0}^{A-1} e^{\frac{2\pi i j r l}{Q}} \right|^2 = \frac{1}{QA} \left| \sum_{j=0}^{A-1} e^{\frac{2\pi i j r l}{Q}} \right|^2$$

weil

$$\left| e^{\frac{2\pi i k_0 l}{Q}} \right|^2 = (\cos^2 \theta + \sin^2 \theta)^2 = 1$$

Example 4.4. Es sei $N = 15$ und $Y = 7$. gesucht wird $\text{Ord}(Y, N) = \min_r \{Y^r \equiv 1 \pmod{N}\}$. Es gilt

- $Y = 7$
- $Y^2 \equiv 4$
- $Y^3 \equiv 13$
- $Y^4 \equiv 1$, d.h. der Algorithmus müsste $r = 4$ finden

Wir wählen jetzt $Q \gg N$, z.B. $Q = 2^8$ und da $N < 2^4$, also $n = 4$ und $q = 8$, beginnen wir mit der Eingabe

$$|0^4\rangle|0^8\rangle \rightarrow \frac{1}{\sqrt{2^8}} \sum_{k=0}^{Q-1} |k\rangle|0^n\rangle$$

Nun berechnen wird die modulare Potenzfunktion auf dem zweiten Qubit und erhalten

$$\frac{1}{\sqrt{2^8}} \sum_{k=0}^{Q-1} |k\rangle|Y^k \pmod{N}\rangle$$

Wir wissen bereits, dass das zweite Qubit daher nur die Werte 7, 4, 13 und 1 annehmen kann. Insbesondere wissen wir $A = \frac{Q}{r} = 64$. Nun wird das zweite Qubit gemessen, erhält man z.B. als Wert $m = 7$, so korrespondiert das zu dem Term

$$\frac{1}{\sqrt{64}} \sum_{j=0}^{63} |jr + 1\rangle|7\rangle$$

Nun wird eine Quanten-Fourier-Transformation auf das erste Qubit angewandt und man erhält

$$\frac{1}{2^8} \sum_{l=0}^{63} \left(\frac{1}{8} \sum_{j=0}^{63} e^{\frac{2\pi i(4j+1)l}{2^8}} |l\rangle|7\rangle \right)$$

Nun wird auf dem ersten Qubit gemessen und wir erhalten den Wert l mit Wahrschein-

lichkeit

$$\mathbb{P}(l) = \frac{1}{QA} \left| \sum_{j=0}^{A-1} \left(e^{\frac{2\pi i r l}{Q}} \right)^j \right|^2$$

Das ist eine geometrische Summe, für die gilt

$$\sum_{j=0}^{A-1} c^j = \frac{1 - c^A}{1 - c}$$

für $|c| < 1$, was in unserem Fall erfüllt ist. Nun gibt es zwei Fälle:

Fall 1: r teilt Q , d.h. $A = \frac{Q}{r}$.

Dann

$$\sum_{j=0}^{A-1} \left(e^{\frac{2\pi i r l}{Q}} \right)^j = \sum_{j=0}^{A-1} \left(e^{\frac{2\pi i l}{A}} \right)^j = \begin{cases} A, & \text{falls } e^{\frac{2\pi i l}{A}} = 1 \text{ und } \frac{l}{A} \in \mathbb{N} \\ \frac{1 - \left(e^{\frac{2\pi i l}{A}} \right)^A}{1 - \left(e^{\frac{2\pi i l}{A}} \right)} = 0 & \text{sonst} \end{cases}$$

Wir messen daher nur solche l mit $\frac{lr}{Q} \in \mathbb{N}$ und alle sind gleich wahrscheinlich. Nun gilt also $l \in \{0, 1, \dots, Q-1\}$, wobei $lr = Qm$ für ein $m \in \mathbb{Q}$. D.h. $lr \in \{0, Q, 2Q, \dots, mQ, \dots, (r-1)Q\}$ induziert eine Gleichverteilung in m .

Tatsächlich wissen wir, dass $r = 4$, also ist die Situation folgende: $l = 2^6 m$ und wegen $l \in \{0, 2^6, 2 \cdot 2^6, 3 \cdot 2^6\}$ gilt $m \in \{0, 1, 2, 3\}$. Ist nun $ggT(m, r) = 1$, so ist r der Nenner von $\frac{l}{Q}$. Nun ist die Frage, mit welcher Wahrscheinlichkeit $ggT(m, r) = 1$ für $m \in \{0, 1, \dots, r-1\}$ gilt.

→ falls m eine Primzahl ist, so gilt trivial $ggT(m, r) = 1$. Da die Anzahl der Primzahlen kleiner n ungefähr durch $\frac{n}{\ln n}$ gegeben ist, gilt

$$\mathbb{P}(ggT(m, r) = 1) \geq \mathbb{P}(m \text{ prim}) \geq \frac{\frac{r}{\ln r}}{r} = \frac{1}{\ln r} \geq \frac{1}{\ln N}$$

Führen wir diesen Algorithmus zum Beispiel $200 \cdot \ln N$ mal aus, so ist der Fehler $< 2^{-100}$.

Fall 2: r teilt nicht Q , d.h. $A = \lceil \frac{Q}{r} \rceil \neq \frac{Q}{r}$.

Falls $lr \bmod Q \approx 0$, oder konkreter, falls

$$-\frac{r}{2} \leq lr \bmod Q \leq \frac{r}{2}$$

Genau r Werte l erfüllen $-\frac{r}{2} \leq l \bmod Q \leq \frac{r}{2}$. Für solche Werte ist

$$e^{\frac{2\pi i l r}{Q}} = e^{\frac{2\pi i (Q \cdot k + m)}{Q}} = e^{\frac{2\pi i m}{Q}}$$

für ein m mit $-\frac{r}{2} \leq m \leq \frac{r}{2}$. Wir schließen, dass

$$\mathbb{P}(\text{Messung} = l) = \frac{1}{QA} \left| \sum_{j=0}^{A-1} \left(e^{\frac{2\pi i m}{Q}} \right)^j \right|^2 \geq \frac{1}{QA} \left| \sum_{j=0}^{A-1} \left(e^{\frac{2\pi i \frac{r}{2}}{Q}} \right)^j \right|^2 = \frac{1}{QA} \left| \frac{1 - e^{\frac{\pi i r A}{Q}}}{1 - e^{\frac{\pi i r}{Q}}} \right|^2$$

An dieser Stelle können die Exponentialdarstellungen in $\cos \theta + i \sin \theta$ umgeschrieben werden, wodurch man

$$\frac{1}{QA} \left(\frac{1 + \cos^2 \theta - 2 \cos \theta + \sin^2 \theta}{1 + \cos^2 \gamma - 2 \cos \gamma + \sin^2 \gamma} \right) = \frac{1}{QA} \frac{1 - \cos \left(\frac{\pi r A}{Q} \right)}{1 - \cos \left(\frac{\pi r}{A} \right)} = \frac{1}{QA} \frac{\sin^2 \frac{\pi r A}{2Q}}{\sin^2 \frac{\pi r}{2Q}} \approx \frac{4Q}{\pi^2 A r^2} \approx \frac{4}{\pi^2 r}$$

Wir schließen, dass die Wahrscheinlichkeit der Messung eines l mit $-\frac{r}{2} \leq l \bmod Q \leq \frac{r}{2}$ mindestens $\frac{4}{\pi^2}$ ist.

Da wir außerdem wissen, dass $|lr - mQ| \leq \frac{r}{2}$ erhalten wir $\left| \frac{l}{Q} - \frac{m}{r} \right| \leq \frac{1}{2Q}$. Darauf kann folgendes Lemma verwendet werden.

Lemma 4.5. *Es gibt höchstens ein Paar (a, b) mit*

$$\left| \frac{l}{Q} - \frac{a}{b} \right| \leq \frac{1}{2Q}$$

mit $b < N$ und $Q \geq N^2$.

Proof. Falls $(a, b), (a', b')$ mit $\frac{a}{b} > \frac{a'}{b'}$ und $b, b' \leq N$, dann ist $\frac{a}{b} - \frac{a'}{b'} = \frac{ab' - a'b}{bb'} \leq \frac{1}{Q}$. Nun gilt $ab' - a'b > 0$ und $bb' < N^2 < Q$, wodurch der Widerspruch gezeigt ist. \square

Für die Darstellung wird die Methode der Continued fractions eingeführt. Für $\alpha \in \mathbb{R}$ ist

$$\alpha = \alpha_0 + \frac{1}{\alpha_1 + \frac{1}{\alpha_2 + \frac{1}{\alpha_3 + \dots}}}$$

. Für diese Darstellung gilt, dass falls $\alpha = \frac{p}{q} \in \mathbb{Q}$, dann konvergiert die CF Darstellung von α in $\mathcal{O}(\max\{|p|, |q|\})$ vielen Schritten. Insbesondere ist $CF_n(\alpha) = \frac{p_n}{q_n}$ im Allgemeinen die beste Approximation von α mit Nenner $\leq q_n$.

Auf das Ordnungsproblem angewandt, finden wir also $CF_k(\frac{1}{Q}) = \frac{p_k}{q_k}$ mit $q_k < N$, also $\frac{p_k}{q_k} = \frac{m}{r}$. Falls nun $\text{ggT}(m, r) = 1$, dann ist $r' = q_k$. Das ist aber nur ein Kandidat für r . An dieser Stelle kann getestet werden, ob $Y^{r'} \equiv 1 \bmod N$. Falls nicht ist $r' \leq r$ und der Algorithmus kann wiederholt werden.

5 Problem der verborgenen Untergruppe

Definition 5.1. Eine Gruppe wird als (G, \circ) notiert. Diese erfüllt Assoziativität, die Existenz eines neutralen Elements und die Existenz eines inversen Elements. Eine Untergruppe $H < G$ ist eine abgeschlossene Teilmenge.

Definition 5.2 (Graph Automorphie). Für einen Graphen $B = (V, E)$ sei G die Menge der Automorphismen auf G . Dabei ist $f : V \rightarrow V$ bijektiv ein Automorphismus, wenn $\forall u, v \in V$ gilt $uv \in E$ genau dann, wenn $f(u)f(v) \in E$. G ist auch eine Gruppe.

Eine Untergruppe H zerlegt G in Nebenklassen $G = \bigcup_{g \in G} gH$ wobei $gH = \{gh : h \in H\}$.

Lemma 5.3. Sei G eine Gruppe und $H < G$. $\forall g_1, g_2 \in G$ gilt $g_1H = g_2H \Leftrightarrow g_1^{-1}g_2 \in H$.

Lemma 5.4. Sei G eine endliche Gruppe, $H < G$. Dann haben alle Nebenklassen genau $|H|$ viele Elemente.

Mit diesen Informationen beschäftigen wir uns nun mit dem Problem der verborgenen Untergruppe.

Gegeben sei eine endliche Gruppe G und $f : G \rightarrow \{0, 1\}^n$ mit einem Black-Box Schaltkreis. f habe die Eigenschaft, dass eine Untergruppe H existiere sodass

$$\forall g \in G \forall h \in H \quad f(g) = f(gh)$$

f ist also konstant in jeder Nebenklasse aber $\forall g_1 \neq g_2$ und $g_1H \neq g_2H$ soll

$$f(g_1) \neq f(g_2)$$

Die Aufgabe ist nun, H zu finden.

Man kann sehen, dass viele bereits bekannte Probleme auf das Problem der verborgenen Untergruppe zurückführbar sind.

Remark 5.5 (Simon). Hier ist gegeben $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ und $\exists a$ sodass $f(x) = f(x \oplus a)$. Man soll a finden.

Sei nun $G = \mathcal{F}_2^n$ mit der bitweisen Operation \oplus . G ist eine Gruppe. Definiert man nun $H = \{0^n, a\}$, so ist H eine Untergruppe und es gilt $f(gh) = f(g)$ für alle $h \in H$. Außerdem ist $G = x_1H \cup x_2H \cup \dots \cup x_NH$ mit $x_i \neq x_j \oplus a$ für alle $i \neq j$. Es handelt sich also um ein Untergruppen Problem, das man lösen kann, wenn H gefunden wird.

Remark 5.6 (Bernstein-Varizani). Gegeben ist $f : \{0, 1\}^n \rightarrow \{0, 1\}$ und es existiert ein a sodass $f(x) = xa$. Die Aufgabe ist, a zu finden.

Wir definieren nun $G = \mathcal{F}_2^n$ wieder mit der bitweisen Paritätsfunktion \oplus . Wir definieren

$$H = \{x \in \{0, 1\}^n | x \perp a\} = \{x \in \{0, 1\}^n | x \cdot a = 0\}$$

H ist eine Untergruppe und da G endlich und $G = x_1 H \cup x_2 H \cup \dots \cup x_N H$ gilt für jedes x_i :

$$f(x_i \oplus h) = (x_i \oplus h)a = x_i \cdot a \oplus \underbrace{h \cdot a}_{=0} = f(x_i)$$

Die andere Eigenschaft ist trivial auch erfüllt und damit ist auch dieses Problem auf das Untergruppenproblem rückführbar.

Remark 5.7 (Shor). Gegeben Y, N mit $ggT(Y, N) = 1$. Finde $\min\{r | Y^r \equiv 1\}$.

Wir definieren $G = (\mathbb{Z}, +)$ mit der Addition als unsere Gruppe. Es sei $H = \{rm | m \in \mathbb{Z}\}$.

Die Nebenklassen von H sind dann $\mathbb{Z} = r\mathbb{Z} \cup 1 + r\mathbb{Z} \cup \dots \cup (r-1) + r\mathbb{Z}$. Außerdem sei $f_N(x) = Y^x \mod N$.

Gilt $x \in a + r\mathbb{Z}$, dann ist $f(x) = f(a + rk) = Y^{a+rk} \mod N = Y^a \mod N \cdot \underbrace{Y^{rk} \mod N}_{=1} = f(a)$. Andererseits für $a \neq b < r$ würde gelten, wenn $f(a) = f(b)$, dass

$$Y^a \mod N = Y^b \mod N$$

und damit wäre $Y^{a-b} = 1 \mod N$. Da man o.B.d.A. von $a > b$ ausgehen kann, gilt, dass $r = a - b < r$, was einen Widerspruch darstellt, oder dass $a = b$, was auch einen Widerspruch darstellt. Also lässt sich auch dieses Problem auf das Untergruppenproblem reduzieren.

Theorem 5.8. Ist G abelsch, dann ist das HSP für G in BQP.

Lemma 5.9. Sei $G = (G\mathbb{F}_2^n, \oplus)$. Sei $H < G$ und $H^\perp = \{h \in G | g \cdot h = 0 \forall g \in G\}$. Für $y \in G\mathbb{F}_2^n$ gilt

$$\sum_{h \in H} (-1)^{hy} = \begin{cases} |H|, & y \in H^\perp \\ 0, & \text{sonst} \end{cases}$$

Wir geben nun einen Algorithmus an, der ein Element aus H^\perp findet.

$$|0^n\rangle|0^n\rangle \xrightarrow{H} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|0^n\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle$$

Sei nun T eine Menge mit einem Element aus jeder Nebenklasse von H . Dann gilt

$$|T| = \frac{|G|}{|H|} = \frac{2^n}{|H|}$$

Es gilt darum weiter

$$\begin{aligned} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle &= \frac{1}{\sqrt{2^n}} \sum_{g \in T} \sum_{x \in H} |g \oplus x\rangle |f(g)\rangle \xrightarrow{H} \frac{1}{\sqrt{2^n}} \sum_{g \in T} \sum_{x \in H} \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{(g \oplus x)y} |y\rangle |f(g)\rangle \\ &= \frac{1}{2^n} \sum_{g \in T} \sum_{y \in \{0,1\}^n} (-1)^{gy} \sum_{x \in H} (-1)^{xy} |y\rangle |f(g)\rangle = \frac{|H|}{2^n} \sum_{g \in T} \sum_{y \in H^\perp} (-1)^{gy} |y\rangle |f(g)\rangle \end{aligned}$$

Nun wird auf dem ersten Qubit gemessen und man erhält $y \in H^\perp$ mit Wahrscheinlichkeit

$$\sum_{g \in T} \left(\frac{|H|}{2^n} (-1)^{gy} \right)^2 = \sum_{g \in T} \frac{|H|^2}{2^{2n}} = \frac{|T| |H|^2}{2^{2n}} = \frac{|H|}{2^n}$$

Insbesondere können alle $y \in H^\perp$ mit der gleichen Wahrscheinlichkeit gemessen werden. Hat H^\perp nun die Dimension d , werden d Vektoren y um

$$\mathbb{P}[\text{alle Vektoren sind linear unabhängig}] \geq \frac{1}{4}$$

5.1 Graphenautomorphie Problem

Wir wissen bereits, dass sich dieses Problem als HSG Problem darstellen lässt. Allerdings unterscheidet sich die Formulierung von den meisten anderen insofern, dass die erzeugte Gruppe nicht abelsch ist.

Formal definiert ist ein Automorphismus eine Abbildung $\phi : V \rightarrow V$ wobei $\forall u, v \in V$ gilt $(u, v) \in E \iff (\phi(u), \phi(v)) \in E$. Die Menge der Automorphismen ist eine Untergruppe der Menge der Permutationen auf V . Insbesondere kann ein $f : \{\text{Permutationen}\} \rightarrow \{0,1\}^n$ definiert werden, sodass $f(\phi) = \phi(G)$. Ist ϕ ein Automorphismus, dann gilt $\phi(G) = G$. Die Fragestellung kann nun formuliert werden als

Gegeben G , existiert ein nicht-trivialer Automorphismus auf $V(G)$?

Dieses Problem ist eng verwandt mit dem Graphisomorphismus Problem, denn G und H sind isomorph $\iff |AUT(G \cup H)| = 2 |AUT(G)| |AUT(H)|$.

6 Endliche Quantenautomaten

Definition 6.1 (endlicher Automat). Ein klassischer Automat, genannt 1DFA ist gegeben als

$$A = (Q, \Sigma, \delta, z_0, E)$$

wobei

- Q die Menge der Zustände

- Σ das Alphabet
- $\delta : Q \times \Sigma \rightarrow Q$ die Überföhrungsfunktion
- z_0 der Anfangszustand
- und $E \subset Q$ die Menge der Endzustände ist

δ kann für ein $a \in \Sigma$ als Matrix beschrieben werden, durch

$$(M_a)_{j,i} = \begin{cases} 1, & \delta(q_i, a) = q_j \\ 0, & \text{sonst} \end{cases}$$

wobei q_i, q_j Zustände in Q seien. Diese können weiter als Vektoren geschrieben werden, wodurch δ mit einer Matrixmultiplikation dargestellt werden kann.

Probabilistische Automaten verallgemeinern die 1DFA dadurch, dass ein Übergang beim Lesen eines Buchstabens $\sigma \in \Sigma$ mehrere Zustände annehmen kann. Jeder Übergang ist einer Wahrscheinlichkeit zugeordnet. In diesem Modell ist ein Zustand wieder ein Vektor, wobei die Komponenten jedem möglichen Zustand eine Wahrscheinlichkeit zuordnen. Ein Zustand für $Q = \{q_1, q_2\}$ der Form $\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$ wäre also mit 50%-iger Wahrscheinlichkeit q_1 und genauso q_2 . Die Übergangsmatrix ist dann auch anders definiert:

Es gilt $\delta(q_i, a, q_j) = \mathbb{P}[q_i \xrightarrow{a} q_j]$. Für $a \in \Sigma$ ist $(M_a)_{j,i} = \delta(q_i, a, q_j)$

Definition 6.2. Ein endlicher Quantenautomat, genannt 1QFA, ist definiert durch ein Tupel

$$A = (Q, \Sigma, \delta, q_0, E_a, E_r)$$

wobei

- Q die Menge der Zustände,
- Σ das Alphabet,
- δ die Übergangsfunktion,
- q_0 der Startzustand,
- E_a die Menge der akzeptierenden Zustände,
- E_r die Menge der verwerfenden Zustände ist.

Insbesondere muss als $E_a, E_r \subset Q$ mit $E_a \cap E_r = \emptyset$ gelten. $E_n = Q \setminus (E_a \cup E_r)$ ist die Menge der Messzustände. Wir definieren die Basiszustände $|q_i\rangle$ mit $q_i \in Q$. Möglich sind aber auch Quantenzustände $|q\rangle = \sum_{q_i \in Q} \alpha_i |q_i\rangle$ mit $\alpha_i \in \mathbb{C}$. Weiter gilt $\delta : Q \times \Sigma \times Q \rightarrow \mathbb{C}$

mit $\delta(q_i, a, q_j)$ ist die Amplitude von q_j im erreichten Quanten-Zustand. Damit ist für $a \in \Sigma$

$$(M_a)_{j,i} = \delta(q_i, a, q_j)$$

und M_a ist unitär. Einen Übergang kann man auch über einen linearen Operator V_a definieren. Dabei ist

$$V_a|q\rangle = \sum_{q' \in Q} \delta(q, a, q')|q'\rangle$$

wobei $\sum_{q' \in Q} |\delta(q, a, q')|^2 = 1$. Dafür wird a aus dem Arbeitsalphabet $\Gamma = \Sigma \cup \{\#, \$\}$. Diese sind als Start- und Endbuchstaben gedacht und sind unabhängig von Σ .

Wir führen nun ein, wie Messungen mit 1QFA funktionieren.

6.1 Messung nach Lesen von jedem Symbol

Es gilt $Q = E_a \cup E_r \cup E_n$. Ein Zustand $|\psi\rangle$ wird gemessen und man erreicht einen von drei Fällen. Ist das Ergebnis der Messung in E_a wird akzeptiert und angehalten, ist es in E_r wird verworfen und angehalten und ist das Ergebnis in E_n , so wird normiert und weiter gemacht.

Sei $|\psi\rangle = \alpha_0|q_0\rangle + \alpha_1|q_1\rangle + \alpha_2|q_2\rangle + \alpha_3|q_3\rangle$ ein konkreter Zustand. Sei weiter $E_a = \{q_0\}$, $E_r = \{q_1\}$, $E_n = \{q_2, q_3\}$. Mit Wahrscheinlichkeit $|\alpha_0|^2$ wird akzeptiert, mit Wahrscheinlichkeit $|\alpha_1|^2$ wird verworfen und mit Wahrscheinlichkeit $|\alpha_2|^2 + |\alpha_3|^2$ wird weiter gerechnet. Der Folgezustand müsste im letzten Fall normiert werden und hat damit die Form

$$(\alpha_2|q_2\rangle + \alpha_3|q_3\rangle) / \sqrt{|\alpha_2|^2 + |\alpha_3|^2}$$

6.2 Beschreibung der Berechnung

Wir betrachten Tupel der Form $(|\psi\rangle, p_a, p_r)$ vom Zustand und den bisherigen Halte-Wahrscheinlichkeiten. Am Anfang ist daher der Zustand $(|q_0\rangle, 0, 0)$. Lesen von $b \in \Sigma$ ist darstellbar als die Überführung zu

$$(T_b|q_0\rangle, p_a, p_r)$$

Definition 6.3. Ein 1QFA A akzeptiert $L \subset \Sigma^*$ mit Wahrscheinlichkeit $\frac{1}{2} + \varepsilon$, wenn

1. $x \in L$ impliziert, $\mathbb{P}(A \text{ akzeptiert}) \geq \frac{1}{2} + \varepsilon$, wobei $x = b_1 b_2 \dots b_n$ und damit

$$T_{\$} T_{b_n} \dots T_{b_2} T_{b_1} T_{\#}(|q_0\rangle, 0, 0) = (|\psi\rangle, p_a, p_r)$$

und damit $p_a \geq \frac{1}{2} + \varepsilon$ gelten muss.

2. $x \notin L$ impliziert $p_r \geq \frac{1}{2} + \varepsilon$.

Theorem 6.4. $1QFA \subsetneq REG$.

Die Frage ist nun, ob man Quantenautomaten stärker machen kann, um alle regulären Sprachen abzudecken.

6.3 Der Two-Way-Quanten-Automat

Wir beginnen zuerst mit dem klassischen Pendant.

Definition 6.5 (2DFA). Ein 2-DFA ist definiert wie ein 1DFA mit

$$\delta : Q \times \Gamma \rightarrow Q \times \{L, N, R\}$$

wobei L, N, R die Richtungen der dazugehörigen Turingmaschine sind. Dieses Modell ist aber nicht stärker als die 1DFA.

Definition 6.6 (2QFA). Sei $Q = \{q_0, \dots, q_n\}$. Im eindimensionalen Fall waren die Basiszustände definiert als $|q_i\rangle$. Im zweidimensionalen Fall sind die Basiszustände definiert durch $|q, i\rangle$ mit $q \in Q$, $i \in \{1, \dots, n\}$. q ist dabei ein Zustand und i eine Position im Eingabestring. Die Übergangsfunktion δ wird definiert durch

$$\delta : Q \times \Gamma \times Q \times \{-1, 0, 1\} \rightarrow \mathbb{C}$$

Die Funktion der 1QFA erfüllt auch die Orthogonalitätseigenschaft

$$\delta(q_1, b, q')\delta(q_2, b, q')^* = \begin{cases} 1, & q_1 = q_2 \\ 0, & q_1 \neq q_2 \end{cases}$$

Die Funktion für 2QFA muss die selbe Eigenschaft erfüllen.

Wir zerlegen die Übergangsfunktion in zwei Teile. Der erste Teil kümmert sich um die Zustände und ist gleich definiert wie im Falle eines 1QFA. Die Richtung wird definiert durch

$$D : Q \rightarrow \{-1, 0, 1\}$$

Im Allgemeinen sei $D(q, i) = (q, i + d \bmod n)$ für $d \in \{-1, 0, 1\}$.

Theorem 6.7. *Ein 2QFA kann nicht reguläre Sprachen erkennen. Es gilt*

$$1QFA \subsetneq REG \subsetneq 2QFA$$

7 Quantenkommunikationskomplexität

7.1 klassische Varianten

Die Komplexität wird hier nicht mehr anhand der Laufzeit gemessen, sondern daran, wie viel Information ausgetauscht werden muss. Wir betrachten die Kommunikation zwischen zwei Parteien A und B. Seien $X, Y \subseteq \{0, 1\}^n$. Es sei $f : X \times Y \rightarrow \{0, 1\}$. Wir nehmen an, A habe die Information x und B habe y . Nun ist die Frage, wie viele Bits ausgetauscht werden müssen, damit A und B f berechnen können.

Example 7.1. 1. Seien $X = Y = \{0, 1\}^n$ und $f(x, y) = \bigoplus_{i=1}^n x_i \oplus \bigoplus_{j=1}^n y_j$. Soll nun A die Lösung kennen, so müsste nur ein Bit von B nach A geschickt werden. Wenn B die Information $\bigoplus_{j=1}^n y_j$ nach A schickt, so kann A f berechnen.

2. Sei $g_2(x, y) = \begin{cases} 1, & |x|_1 = |y|_1 \\ 0, & \text{sonst} \end{cases}$. Im Worst-Case sind nun alle $y_j = 1$. Dann müsste B $\lceil \log(|y|_1) \rceil$ Bits zu A schicken.

3. Sei $f_3(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{sonst} \end{cases}$. In diesem Fall sind n Bits ausreichend und notwendig. Wir zeigen wir nun, dass n Bits notwendig sind?

Es muss ein Kommunikationsprotokoll vorab vereinbaren, das beide Parteien kennen. Dieses Protokoll lässt sich im Allgemeinen als Spielbaum darstellen, wobei die Baumtiefe der Menge an zu versendenden Bits entspricht. Außerdem gibt es $\forall (x, y) \in X \times Y$ einen Pfad im Baum. Daher kann man sehen, dass falls die Kommunikationskomplexität $\leq n - 1$ ist, dann $\exists w_1 \neq w_2 \in X = Y$ mit $\text{Pfad}(w_1 w_1) = \text{Pfad}(w_2 w_2) = \text{Pfad}(w_1 w_2)$, da der Teilpfad bis zum Ende von w_1 dem Teilpfad bis zum Ende von w_2 entspricht. Das geht weiter, wenn anschließend das Wort w_2 gelesen wird. Das ist aber ein Widerspruch, da $w_1 \neq w_2$ und daher $\text{Pfad}(w_1 w_2)$ in einer 0 endet, die anderen beiden Pfade hingegen in einer 1.

7.2 probabilistische Varianten

Hier sind wir auf der Suche nach Protokollen, sodass A berechnet $f(x, y)$ mit Fehlerwahrscheinlichkeit $\leq \varepsilon$. Für ein Problem P bezeichnen wir mit $C_\varepsilon(P)$ die Kommunikationskomplexität von P mit Fehlerwahrscheinlichkeit $\leq \varepsilon$, wobei $C(P) = C_0(P)$ die deterministischen Kommunikationskomplexität.

Example 7.2. Wir arbeiten weiter mit der Funktion $f_3 = EQ$ vom obigen Beispiel und möchten die Fehlerwahrscheinlichkeit $\leq \varepsilon$ halten. Dafür lassen wir A und B die kleinste

Primzahl p berechnen, sodass $p > \frac{n}{\varepsilon}$. Die folgenden Berechnungen finden statt in \mathbb{Z}_p . B bestimmt mit den Bits von y ein Polynom q der Form

$$q(z) = y_1 + y_2 z + y_3 z^2 + \dots + y_n z^{n-1}$$

und wählt ein zufälliges $v \in_R \mathbb{Z}_p$ und berechnet $q(v) = w$. Anschließend werden v und w an A geschickt. A berechnet daraufhin das Polynom

$$P(z) = x_1 + x_2 z + x_3 z^2 + \dots + x_n z^{n-1}$$

Wenn $P(v) = w$, dann wird 1 als Antwort zurückgegeben, andernfalls 0. Aus der Zahlentheorie wissen wir, dass $p \leq \frac{2n}{\varepsilon}$, weswegen der Algorithmus nur $2\lceil \log(\frac{2n}{\varepsilon}) \rceil = \mathcal{O}(\log n)$ Bits verschickt. Es fehlt nun noch zu zeigen, dass die Fehlerwahrscheinlichkeit in der Tat $\leq \varepsilon$ ist. Wenn $P(v) \neq w$ ist, dann ist die Antwort 0 gerechtfertigt, da $x \neq y$ sein müssen. Wenn allerdings 1 zurückgegeben wird, dann könnte es trotzdem sein, dass $x \neq y$ obwohl $P(v) = q(v)$. Es gilt aber

$$\mathbb{P}_v[q(v) = P(v) | q \neq P] = \mathbb{P}_v[q(v) - P(v) = 0 | q \neq P] = \mathbb{P}_v[v \text{ ist NST von } q - P]$$

Der Grad von $q - P$ ist höchstens $n - 1$ und daher ist die Anzahl der Nullstellen $\leq n - 1$. Daher ist obige Wahrscheinlichkeit

$$\leq \frac{n - 1}{p} \leq \frac{n}{\frac{n}{\varepsilon}} \leq \varepsilon$$

7.3 Quantenkommunikationskomplexität

Für Quantenkommunikation gibt es zwei verschiedene Varianten:

Definition 7.3. • A und B schicken sich klassische Bits und dürfen nur vor dem Spiel Quantenzustände kommunizieren und verschränken

- Die Spieler dürfen Qbits übertragen

Man kann zeigen, dass beide Modelle gleichmächtig sind. Wie beginnen mit einem Beispiel für die zweite Variante.

Example 7.4. Diesmal betrachten wir drei Spieler A, B, C mit $x_A, x_B, x_C \in \{0, 1\}$, sodass

$$x_A \oplus x_B \oplus x_C = 0$$

Die Aufgabe an die drei Spieler ist, drei Bits a, b, c zu produzieren, sodass

$$a \oplus b \oplus c = x_A \vee x_B \vee x_C$$

Klassisch ist das nicht möglich zu lösen. Wir bezeichnen (für alle Spieler, aber am Beispiel von A) a_i als das Bit, das A produziert, falls $x_A = i \in \{0, 1\}$. Es gibt nun vier Fälle

$$a_0 \oplus b_0 \oplus c_0 = 0$$

$$a_0 \oplus b_1 \oplus c_1 = 1$$

$$a_1 \oplus b_0 \oplus c_1 = 1$$

$$a_1 \oplus b_1 \oplus c_0 = 1$$

Hieran sieht man, dass dieses Problem deterministisch nicht lösbar ist. Auch eine probabilistische Lösung kann keine niedrige Fehlerwahrscheinlichkeit erreichen, da immer nur 3 der obigen Gleichungen gleichzeitig wahr sein können. Wir betrachten daher im Folgenden die Quantenlösung.

Die Spieler teilen initial das Qbit

$$|\phi\rangle = \frac{-1}{2}(|000\rangle - |011\rangle - |101\rangle - |110\rangle)$$

den sogenannten GHZ Zustand. Jeder der Spieler (am Beispiel von A) führt anschließend folgendes Protokoll aus: Falls $x_A = 0$, Messung von $\langle \phi \rangle$ bezüglich der Standardbasis. Falls $x_A = 1$, eine Messung bezüglich der Basis $\{|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\}$. Anschließend setzt A den Wert a auf das Ergebnis der Messung. Nun gibt es zwei Fälle

1. Fall: $(x_A, x_B, x_C) = (0, 0, 0)$. In diesem Fall ist es egal, welches Ergebnis die Messung zurückgibt, denn die Summe ist immer 0.
2. Fall: $(x_A, x_B, x_C) \neq (0, 0, 0)$. In diesem Fall messen zwei der Spieler bezüglich der Dualbasis. Wenn man $|\phi\rangle$ bezüglich dieser Basis umschreibt, so erhält man

$$|\phi\rangle = \frac{1}{4}(|+-0\rangle - |-+0\rangle - |+ +1\rangle + |- -1\rangle)$$

wodurch als Ergebnis nur $(0, 1, 0), (1, 0, 0), (0, 0, 1)$ oder $(1, 1, 1)$ zurückgegeben werden kann. Dieser erfüllen alle die Bedingung $a \oplus b \oplus c = x_A \vee x_B \vee x_C$.

Example 7.5. Wir betrachten nun ein zweites Beispiel mit drei Spielern A, B, C und den Worten $x = x_1x_0, y = y_1y_0, z = z_1z_0$, wobei $x + y + z = 0 \pmod{2}$, $x_0 \oplus y_0 \oplus z_0 = 0$ und

der zu berechnenden Funktion

$$f(x, y, z) = \frac{x + y + z \bmod 4}{2}$$

d.h.

$$f(x, y, z) = \begin{cases} 0, & x + y + z = 0 \\ 1, & x + y + z = 2 \end{cases}$$

Offensichtlich gilt $f(x, y, z) = x_1 \oplus y_1 \oplus z_1 \oplus (x_0 \oplus y_0 \oplus z_0)$. Alle drei Spieler sollen f berechnen. Mit Quantenkommunikation sind 3 Bits notwendig, wobei ein deterministischer Ansatz 4 Bits benötigt.

Definition 7.6. Wir geben einige Kommunikationsprobleme an, die später nützlich sein werden.

1. $Disj(x, y) = \bigvee_{i=1}^n x_i \wedge y_i$. Hier gilt $C_\varepsilon(Disj) = \Omega(n)$ und $C_0(Disj) = \Omega(n)$

$$2. EQ'(x, y) = \begin{cases} 1, & x = y \\ 0, & \Delta(x, y) = \frac{n}{2} \text{ wobei } \Delta \text{ der Hamming Abstand sei.} \\ NaN, & \text{sonst} \end{cases}$$

Für dieses Problem gilt $C_\varepsilon(EQ') = \mathcal{O}(1)$, wenn gemeinsame Zufallsbits verwendet werden können. Wenn die Zufallsbits von A und B unabhängig sind, ist die Kommunikationskomplexität $\mathcal{O}(\log n)$. Es gilt immer $C_0(EQ') = \Omega(n)$.

Definition 7.7. Für das Folgende ist ein bisschen Notation nötig.

$$\mathcal{F}_n = \{f : \{0, 1\}^n \rightarrow \{0, 1\}\}$$

Beachte, dass man anhand der Wahrheitstafel jedes f als eine 2^n -lange Bitfolge definieren kann. Da es $2^n = N$ viele solche f gibt, ist

$$\mathcal{F}_n \hat{=} \{0, 1\}^N$$

Auf diese Weise kann $L : \{0, 1\}^2 \rightarrow \{0, 1\}$ ausgebaut werden zu

$$L(g, h) : \{0, 1\}^n \rightarrow \{0, 1\}$$

durch

$$\forall x \in \{0, 1\}^n \quad L(g, h)(x) = L(g(x), h(x)) \quad \forall g, h \in \mathcal{F}$$

Theorem 7.8. Sei $F : \mathcal{F}_n \rightarrow \{0, 1\}$ und $L : \{0, 1\}^2 \rightarrow \{0, 1\}$. Falls es einen Quanten-Algorithmus für $F(f)$ mit t Fragen an f der Form

$$|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$$

gibt, dann gibt es ein Quantenkommunikationsprotokoll mit $t(2n+4)$ Qubits für folgendes Problem:

A erhält g , B erhält h und A muss $F(L(g, h))$ berechnen.

Example 7.9. Sei $F = OR_N$ und $L = AND_2$. Was ist $F(L(g, h))$? Es gilt

$$F(L(g, h)) = 1 \Leftrightarrow \exists x \in \{0, 1\}^n : L(g, h)(x) = 1$$

Das wiederum ist genau dann der Fall, wenn es ein x gibt, sodass $g(x) \wedge h(x) = 1$. Man sieht, dass diese Funktion äquivalent zum Disj Problem ist, $F(L(g, h)) = Disj_N(g, h)$. Der Algorithmus von Grover kann auf natürliche Weise verwendet werden, um OR_N zu berechnen (denn man sucht nach einer 1). Das bedeutet, es gibt einen Quantenalgorithmus für $F(f)$ mit $\mathcal{O}(\sqrt{N})$ Fragen. Aus obigem Theorem schließen wir, dass es ein Quantenkommunikationsprotokoll mit $\mathcal{O}(\sqrt{N} \log N)$ Bits gibt.

Example 7.10. Sei $F = OR'$, das heißt

$$F(f) = \begin{cases} 1, & |\{x : f(x) = 0\}| = |\{x : f(x) = 1\}| \\ 0, & f(x) = 0 \forall x \\ NaN, & \text{sonst} \end{cases}$$

und sei $L = \oplus$. Wir stellen uns wieder die Frage, was $F(L(g, h))$ ist.

$$F(L(g, h)) = 0 \Leftrightarrow g(x) \oplus h(x) = 0 \forall x \Leftrightarrow g(x) = h(x) \forall x \Leftrightarrow \Delta(g, h) = 0$$

Auf der anderen Seite ist

$$\Delta(g, h) = \frac{N}{2} \Leftrightarrow |\{x : L(g, h)(x) = 0\}| = |\{x : L(g, h)(x) = 1\}|$$

Es folgt daraus $F(L(g, h)) = \neg EQ'$, denn

$$F(L(g, h)) = \begin{cases} 1, & \Delta(g, h) = 2^{n-1} \\ 0, & \Delta(g, h) = 0 \\ NaN, & \text{sonst} \end{cases}$$

Da die Komplexität für das negierte Problem aber gleich bleibt, können wir aus einer Lösung für $F(L(g, h))$ auch auf EQ' schließen. Der Algorithmus von Deutsch-Josza berechnet genau F mit $t = 1$. Hier wird anstatt NaN aber ein zufälliges Bit zurückgegeben, was aber zu keinen Problemen führt. Wir folgern, dass wir für $\neg EQ'$ (und damit für EQ') eine Quantenkommunikationskomplexität von $\mathcal{O}(\log N)$ erreichen.

7.4 Finger Printing

Finger Printing ist eine Verallgemeinerung von Hashing. Es wird eine große Struktur mittels einer Funktion auf eine viel kleinere Menge abgebildet. Wir wollen Finger Printing verwenden, um nochmals die Kommunikation für EQ zu lösen. Diesmal habe A das Wort x und B das Wort y . Beide sollen nun gewisse Bits zu einem dritten Spieler C schicken, der entscheidet, ob $x = y$. Im deterministischen Fall bleibt eine Wort-Case Komplexität von $\mathcal{O}(n)$. Wir wissen bereits, dass probabilistisch $\mathcal{O}(\log n)$ Bits genügen, wenn gemeinsame Zufallsbits erlaubt sind. Wir führen eine Art von Code anhand seiner Eigenschaften ein, aber geben keine Implementierung an. Mit diesem Code lässt sich das EQ Problem noch effizienter lösen.

Definition 7.11 (Justesen Codes). Der Code ist gegeben durch eine Funktion

$$E : \{0, 1\}^n \rightarrow \{0, 1\}^m$$

wobei $m = cn$ für ein $c > 2$. Der Code erfüllt die Eigenschaften

1. Ist $x = y$, dann $E(x) = E(y)$.
2. Ist $x \neq y$, dann $\Delta(E(x), E(y)) \geq (1 - \delta)m$ für ein $\delta \in (0, \frac{9}{10} + \frac{1}{15c})$ wobei Δ der Hammingabstand ist.

Mit diesem Code definieren wir nun ein Protokoll für die Funktion EQ.

A und B haben Zugriff auf eine Zufallsfolge $i \in \{1, \dots, m\}$. Die Startzustände sind $x, y \in \{0, 1\}^n$. A und B berechnen $E(x)$ bzw. $E(y)$ und wählen aus der Zufallsfolge ein gemeinsamen Index $i \in \{1, \dots, m\}$. Nun schicken beide Parteien das Bit $(E(x))_i$ bzw. $(E(y))_i$ an C. Sind die Werte gleich, gibt C 1 aus und 0 sonst.

Das Protokoll gibt auf jeden Fall richtig 0 aus, denn wenn $E(x) \neq E(y)$, dann ist auch $x \neq y$. Die Fehlerwahrscheinlichkeit ist demnach

$$\mathbb{P}[(E(x))_i = (E(y))_i | x \neq y] = \frac{\delta m}{m} = \delta$$

Wenn man nun den Fehler auf $\leq \varepsilon$ bekommen möchte, so kann man dieses Protokoll k -mal wiederholen. Soll dann gelten, dass $\delta^k \leq \varepsilon$, so erhält man eine Worst-Case Komplexität

von $\mathcal{O}(\log \frac{1}{\varepsilon})$. Das Problem dieser Vorgehensweise liegt im Austausch der gemeinsamen Zufallsfolge. Wenn man diese vermeiden möchte, kann auf einen Quantenansatz zurückgegriffen werden.

Hierbei ist die Idee, die Vektoren x, y zu betrachten und mit einer Funktion f den Zusammenhang $x \neq y \implies \langle f(x)|f(y) \rangle \approx 0$ zu erzeugen. Wie wir sehen werden, ist der Justesen Code eine gute Wahl für f . Durch E erhalten wir also eine Transformation der Form

$$x \mapsto |h_x\rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^m |i\rangle |E(i)\rangle$$

Nach den Eigenschaften von E folgt

- $x = y \implies |h_x\rangle = |h_y\rangle \implies \langle h_x|h_y\rangle = 1$
- $x \neq y \implies \langle h_x|h_y\rangle \leq \delta$.

Wir halten fest, dass $|h_x\rangle$ und $|h_y\rangle$ Quantenzustände der Länge $\log m + 1 = \mathcal{O}(\log n)$ Qubits sind. Beide Zustände werden nun an C geschickt, welcher mit einem dafür ausgelegten Schaltkreis bestimmt, ob $x = y$. Der Schaltkreis hat die drei Eingaben $|0^n\rangle$, $|\phi\rangle = |h_x\rangle$, $|\psi\rangle = |h_y\rangle$. Mit einer Hadamard Transformation auf den ersten n Bits ergibt sich der Zustand

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\phi\rangle|\psi\rangle$$

Anschließend wird ein kontrollierter Swap auf den letzten beiden Qubits mit dem ersten Qubit als Kontrollbit durchgeführt. Wir erreichen den Zustand

$$\frac{1}{\sqrt{2}}(|0\rangle|\phi\rangle|\psi\rangle + |1\rangle|\psi\rangle|\phi\rangle)$$

Auf den ersten n Bits wird wieder eine Hadamard Transformation durchgeführt und wir erhalten den Zustand

$$\frac{1}{2}(|0\rangle(|\phi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle) + |1\rangle(|\psi\rangle|\phi\rangle - |\phi\rangle|\psi\rangle))$$

Wir schreiben nun

$$|\phi\rangle = \sum_i a_i |i\rangle, \quad |\psi\rangle = \sum_i b_i |i\rangle$$

Dadurch lässt sich der letzte Teil schreiben als

$$|\psi\rangle|\phi\rangle - |\phi\rangle|\psi\rangle = \sum_{i,j} (a_i b_j - a_j b_i) |i\rangle |j\rangle$$

Die Wahrscheinlichkeit 1 zu messen, ist demnach

$$\frac{1}{4} \sum_{i,j} |a_i b_j - a_j b_i|^2 = \frac{1}{2} (1 - \langle \phi | \psi \rangle^2)$$

Ist $x = y$, dann ist dieser Ausdruck 0 und man misst auf jeden Fall den Wert 0, ansonsten wird mit positiver Wahrscheinlichkeit 1 gemessen. Wenn eine Fehlerwahrscheinlichkeit von ε erreicht werden soll, so muss dieses Vorgehen $\mathcal{O}(\log \frac{1}{\varepsilon})$ mal wiederholt werden.

7.5 Quantenkryptologie

Der Sender A (Alice) will eine Nachricht an den Empfänger B (Bob) schicken. Die Nachricht bezeichnen wir im Folgenden mit m . A hat eine Verschlüsselungsfunktion f und B hat eine andere Funktion g . Beide haben Zugriff auf einen Schlüssel k . Wenn $g = f^{-1}$, ist das Problem gelöst.

Der Schlüssel ist der Knackpunkt. Ein Beispiel für den Schlüssel k ist zum Beispiel der sogenannte One-Time-Pad. Dabei ist k eine Bitfolge und A berechnet die Funktion $f(m, k) = m \oplus k$, das Bitweise XOR. Anschließend berechnet B die Funktion $f(f(k, m)) = m$. Das Verfahren ist sicher, solange die Übertragung des Schlüssels nicht abgehört wurde. Dieses Problem lässt sich aber mit einem Quantenansatz lösen. Das Protokoll basiert darauf, dass A und B bemerken können, wenn die Information in k abgehört wurde.

Sei $|m| = n$. A führt zuerst die folgenden Schritte aus

1. wähle $a, b \in_R \{0, 1\}^{4n}$
2. for $i = 1$ to $4n$ do
3. $|q_i\rangle = \begin{cases} |a_i\rangle, & b_i = 0 \\ H|a_i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{a_i}|1\rangle), & b_i = 1 \end{cases}$
4. endfor
5. schicke $|q_1\rangle \dots |q_{4n}\rangle$

Anschließend folgt B dem folgenden Protokoll

1. wähle $c \in \{0, 1\}^{4n}$
2. for $i = 1$ to $4n$ do
3. falls $c_i = 0 \Rightarrow$ Messung von $|q_i\rangle$ bezüglich Standardbasis, sonst Messung bezüglich Hadamard Basis.

Anschließend schickt A b an B und B schickt c an A.

Sei $G = \{i | c_i = b_i\}$. Es gilt $\mathbb{E}[|G|] = 2n$. A und B können G bestimmen. Nun wählt A eine Untermenge $G' \subset G$ mit $|G'| = n$ und schickt die Menge der Paare $\{(a_i, i) | i \in G'\}$ an B. B schickt die Menge der Paare $\{(q_i, i) | i \in G'\}$ an A. Ist $a_i = q_i$ für alle $i \in G'$, dann gilt die Verbindung als sicher und es kann vorgegangen werden wie oben. Gibt es ein $i \in G'$ mit $a_i \neq q_i$, so wurde die Kommunikation abgehört.

Wenn eine dritte Person C nun die Kommunikation abgehört hätte, dann müsste sie die $|q_i\rangle$ messen. Da C aber die zufällige Bitfolge c nicht kennt, weiß C nicht nach welcher Basis gemessen werden muss. Für jedes Bit ergibt sich daher die Wahrscheinlichkeit $\frac{1}{2}$, dass die richtige Basis für jedes Bit gewählt wurde und damit die Wahrscheinlichkeit, dass C abhören kann, ohne bemerkt zu werden $(\frac{1}{2})^n$.

8 Quanten fehlerkorrigierende Codes

8.1 klassische fehlerkorrigierende Codes

Die Idee von fehlerkorrigierenden Codes ist, dass man ein Codewort c eines Codes C versendet, und dieses Codewort durch etwaige Störungen in der Übertragung verändert werden kann. Ist C fehlerkorrigierend, so kann das übertragene Wort wieder zurück zu c überführen solange die Anzahl der Fehler nicht zu groß ist. Dieser Rückschluss auf das eigentliche Codewort geschieht zum Beispiel nach dem Maximum Likelihood Prinzip. Die wahrscheinlichsten Codes sind der Form $C = \{0^n, 1^n\}$ (genannt Wiederholungscodes). Diese sind $\lfloor \frac{n-1}{2} \rfloor$ -fehlerkorrigierend. Ein anderes Beispiel sind Paritätscodes C , wobei für $c \in C$ immer gelten muss, dass $|c|_1$ ist gerade. Diese können keinen Fehler korrigieren.

Beide obigen Beispiele sind Subklassen der Menge der linearen Codes. Für einen linearen Code C gilt für alle c, c' , dass $c \oplus c' \in C$ gilt, das heißt C ist ein Unterraum des Raumes der Vektoren. Insbesondere gibt es eine Basis $B = \{b_1, \dots, b_n\}$ für C . Die Matrix M mit den b_i in der i -ten Zeile heißt Generatormatrix für C . Das heißt für alle $w \in \{0, 1\}^n$ ist $M \cdot w$ ein Codewort in C . Eine Prüfmatrix P ist eine boole'sche Matrix, sodass die Spalten eine Basis für C^\perp definieren. Insbesondere gilt $c \in C \iff cP = 0$.

8.2 Quantencodes

Wir unterscheiden zwei verschiedene Fehler, die in einer Quantencode-Übertragung auftreten können:

- Bit-Flips: $a|0\rangle + b|1\rangle$ wird übertragen als $b|0\rangle + a|1\rangle$

- Phase-Flips: $a|0\rangle + b|1\rangle$ wird übertragen als $a|0\rangle - b|1\rangle$

8.2.1 Bit-Flips

Mit einem einfachen Schaltkreis lassen sich die Transformationen $|0\rangle \rightarrow |000\rangle$ und $|1\rangle \rightarrow |111\rangle$ umsetzen. Damit erhält man $\phi \rightarrow \hat{\phi} = a|000\rangle + b|111\rangle$. Würde man nun nach einer fehlerhaften Übertragung $a|001\rangle + b|110\rangle$, so kann man mit dem Maximum-Likelihood-Prinzip wiederum das ursprüngliche Codewort rekonstruieren. Mit einem extra Register, kann man einen Schaltkreis erstellen, der die Positionen der Fehler ausgibt. Mithilfe eines NOT-Gatters auf diesen Bits können diese Fehler schließlich behoben werden.

8.2.2 Phase-Flip

Man bemerke, dass ein Phase-Flip nichts anderes ist, als ein Bit-Flip in der Dualbasis. Man transformiere das Fehlerhaft übertragene Qubit $a|0\rangle - b|1\rangle$ mit einer Hadamard Transformation und erhält

$$\frac{1}{\sqrt{2}}(a - b)|0\rangle + (a + b)|1\rangle$$

Wenn man darauf wiederum einen Bit-Flipm ausführt, so erhält man

$$\frac{1}{\sqrt{2}}(a + b)|0\rangle + (a - b)|1\rangle$$

Nun führe man wiederum eine Hadamard Transformation aus und man erhält dadurch $a|0\rangle + b|1\rangle$ zurück. Es ist nun klar, dass man den Schaltkreis von oben auf Phase-Flips anpassen kann.

8.2.3 9-Bit Code

Die Idee in diesem Code ist es, beide obigen Strategien kombiniert werden. Es wird zunächst ein Bit in neun Versionen davon kopiert. Der Code kann Fehler in irgendeinem der Qubits korrigieren.

8.2.4 CSS Codes

Sei H ein Unterraum von $\{0, 1\}^n$. Die Codewörter sind Überlagerungen von den Vektoren in H . Wir definieren

$$|H\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |h\rangle$$

Bit-Flips können mit einem $\{0, 1\}$ -Vektor g repräsentiert werden. Es gilt $g_i = 1 \iff$ an Stelle i ist ein Fehler. Es gilt

$$|H + g\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |h + g\rangle$$

Für Phase-Flips verwenden wir einen Vektor g' wobei

$$|H + g'\rangle = \sum_{h \in H} (-1)^{g' \cdot h} |h\rangle$$

Es folgt schließlich, wenn beide Fehler betrachtet werden sollen, dann ist

$$|H + g + g'\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} (-1)^{g' \cdot h} |h + g\rangle$$