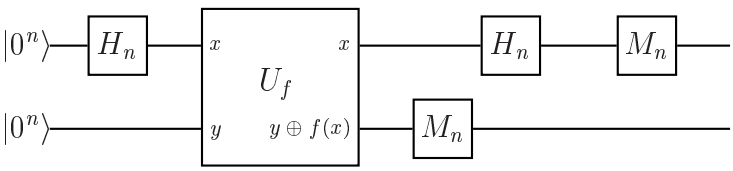
Vorlesungsskript zur Vorlesung

Quantencomputer

Jacobo Torán



Abteilung Theoretische Informatik
Universität Ulm
Sommersemester 2003
Mitschrift von Jan Paul Buchwald

Inhaltsverzeichnis

1 Einführung **5**

1.1 Qubits 5
1.2 Geometrische Repräsentation 6
1.3 Alternative Basen für Qubits 7
1.4 Qubit Messung 7
1.5 Operationen 8
1.6 2-Qubits 9
1.7 Transformationen 10
1.8 n-Qubit-Register 11

2 Quantenschaltkreise **13**

2.1 Reversibilität 13
2.2 Simulation klassischer Schaltkreise 13
2.3 Simulation probabilistischer Schaltkreise 16
2.4 Die Komplexitätsklasse BQP 16

3 Erste Quantenalgorithmen **21**

3.1 Quanten-Parallelismus 21
3.2 Der Deutsch-Algorithmus 22
3.3 Deutsch-Josza-Problem 22
3.4 Simon's Algorithmus 24

4 Suchalgorithmus von Grover **29**

4.1 Die Problematik 29
4.2 Lösung mittels Quantenschaltkreis 30
4.3 Eine untere Schranke für das Problem 32
4.4 Geometrische Interpretation 35
4.5 Anwendungen des Algorithmus 38
4.5.1 Das Minimum-Problem 39

4.5.2 Element distinctness 43

5 Algorithmus von Shor für die Faktorisierung **45**
5.1 Diskrete Fourier-Transformation 45
5.2 Fast Fourier Transformation 46
5.3 Quantum Fourier Transformation 48
5.4 Der Algorithmus für die Faktorisierung 51
5.5 Der Algorithmus zum Ordnungsproblem 52

6 Endliche Quanten-Automaten **59**
6.1 Der One Way Quantum Finite Automaton (1QFA) 59
6.2 Der Two-Way Quantum Finite Automaton (2QFA) 65

7 Quanten-Kommunikationskomplexität und Kryptologie **71**
7.1 Klassische und probabilistische Varianten 71
7.2 Quanten-Kommunikationskomplexität 73
7.3 Finger Printing 79
7.4 Quantenkryptologie 82

Literaturverzeichnis **85**

Kapitel 1

Einführung

1.1 Qubits

Die vollständige Beschreibung eines physikalischen Systems wird als *Zustand* bezeichnet. In der Quantenmechanik wird ein solcher Zustand mit Hilfe eines Vektors der Norm 1 im *Hilbertraum* dargestellt. Die Vektoren werden dabei in einer speziellen Notation, der Dirac-Notation, aufgeschrieben (z.B. $|\psi\rangle$, $|\varphi\rangle$). Falls $|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \dots \\ \alpha_n \end{pmatrix}$ ein Vektor in $\mathbb{C}^n$ ist, dann ist $\langle\psi| = (\overline{\alpha_1}, \dots, \overline{\alpha_n})$, wobei $\overline{\alpha_i}$ die Konjugierte von α_i ist. Der Hilbertraum stellt einen Vektorraum V über $\mathbb{C}$ mit einem inneren Produkt dar, das wie folgt definiert ist:

Definition 1.1.1 *Inneres Produkt* $\langle\psi|\varphi\rangle : V^2 \to \mathbb{C}$ *mit*

$i)$ $\langle\psi|\psi\rangle \geq 0$ *(gleich* 0 *g.d.w.* $|\psi\rangle = 0$*).*

$ii)$ $\langle\psi|(a|\varphi_1\rangle + b|\varphi_2\rangle) = a\langle\psi|\varphi_1\rangle + b\langle\psi|\varphi_2\rangle)$.

$iii)$ $\langle\varphi|\psi\rangle = \langle\psi|\varphi\rangle^*$.

$\mathbb{C}^n$ hat das innere Produkt

$$\langle\psi|\phi\rangle = \sum_{i=1}^{n} \overline{\alpha_i}\beta_i \text{ (für } |\psi\rangle = \begin{pmatrix} \alpha_1 \\ \dots \\ \alpha_n \end{pmatrix} \text{ und } |\phi\rangle = \begin{pmatrix} \beta_1 \\ \dots \\ \beta_n \end{pmatrix})$$

Manchmal, um die Notation zu vereinfachen, werden wir das Produkt $\langle\varphi|\psi\rangle$ auch $(|\varphi\rangle, |\psi\rangle)$ notieren.

Norm: $V \to \mathbb{C} \,:\, \| \, |\varphi\rangle \, \| = \sqrt{\langle\varphi|\varphi\rangle}$.

$|\varphi\rangle$ ist ein *unitärer Vektor* falls $\| \, |\varphi\rangle \, \| = 1$.

Standard-Basis: $B = \{|0\rangle, |1\rangle\}$, wobei $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Ein solcher Vektor $|\psi\rangle$ entspricht einem *Qubit*. Meistens wird ein Qubit mit Hilfe der orthonormalen Basis B dargestellt. Somit lässt sich ein Qubit in der Form $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ mit $a, b \in \mathbb{C}$ notieren. Für die Koeffizienten a und b, die man auch *Amplituden* nennt, muss wegen der Normierung gelten, dass $|a|^2 + |b|^2 = 1$. Ein Qubit $|\psi\rangle$ kann man sich also als eine Superposition von 0 und 1 vorstellen, wobei mit Wahrscheinlichkeit $|a|^2$ gilt, dass $|\psi\rangle = 0$, und mit Wahrscheinlichkeit $|b|^2$, dass $|\psi\rangle = 1$. In der Quantenmechanik lassen sich damit beispielsweise die Polarisierung eines Photons oder die Energiestufen eines Wasserstoffelektrons modellieren. Durch die unendliche Auswahl an Möglichkeiten für die Amplituden a und b kann ein Qubit theoretisch Information von unendlicher Größe speichern, die aber bei jeder Messung wieder auf ein Bit reduziert wird.

1.2 Geometrische Repräsentation

Die komplexen Amplituden a und b können in der komplexen Zahlenebene jeweils durch die zwei Größen Radius und Winkel charakterisiert werden, also etwa $a = (\cos\varphi_a + i\,\sin\varphi_a)\,r_a$ und $b = (\cos\varphi_b + i\,\sin\varphi_b)\,r_b$. Diese Ausdrücke können mit Hilfe der Eulerschen Formel ($e^{i\theta} = \cos\theta + i\sin\theta$) umgeformt werden zu $a = r_a\,e^{i\varphi_a}$ bzw. $b = r_b\,e^{i\varphi_b}$. Damit gilt

$$|\psi\rangle = r_a\,e^{i\varphi_a}\,|0\rangle + r_b\,e^{i\varphi_b}\,|1\rangle$$

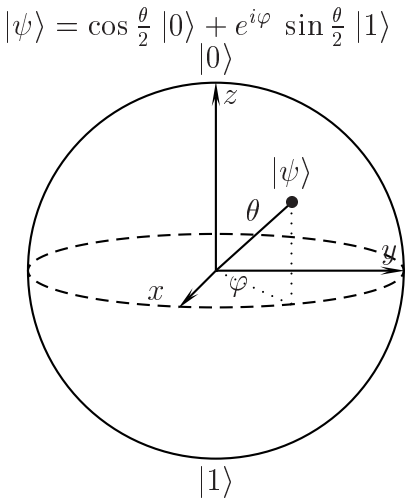
und somit für die Radien

$$r_a^2 + r_b^2 = 1$$

Nimmt man nun eine weitere Dimension zur Beschreibung hinzu, so stellt man aufgrund der letzten Beziehung zwischen den Radien fest, dass diese sich auch durch $r_a = \cos\frac{\theta}{2}$ und $r_b = \sin\frac{\theta}{2}$ für einen Winkel θ schreiben lassen. Insgesamt erhält man dann

$$|\psi\rangle = e^{i\varphi_a}(\cos\tfrac{\theta}{2}\,|0\rangle + e^{i(\varphi_b - \varphi_a)}\,\sin\tfrac{\theta}{2}\,|1\rangle)$$

Da der Term $e^{i\varphi_a}$ aus physikalischen Gründen vernachlässigt werden kann und die Differenz $\varphi_b - \varphi_a$ einen neuen Winkel φ darstellt, kann das Qubit schließlich mit Hilfe folgender Gleichung dargestellt werden:



Bloch-Sphäre

1.3 Alternative Basen für Qubits

Dualbasis B': Es gibt noch eine ganze Reihe weiterer Basen für Qubit-Darstellungen, die häufigste davon ist die Dualbasis mit den Elementen

$$|0'\rangle = |\nearrow\rangle = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$|1'\rangle = |\searrow\rangle = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Wie sich leicht nachvollziehen lässt, gilt $\langle 0'|1'\rangle = 0$ und $\langle 0'|0'\rangle = 1$, d.h. die Dualbasis ist ebenfalls orthonormal. Andere Basen bringen auch komplexe Elemente mit ins Spiel, wie z.B. folgende:

$$|0''\rangle = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix}$$
$$|1''\rangle = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix}$$

1.4 Qubit Messung

Das Ergebnis der Messung eines Qubits ist ein klassisches Informationsbit. Die Messung von $|\varphi\rangle = \alpha|0\rangle + |\beta|1\rangle$ ergibt 0 oder 1 mit Wahrscheinlichkeit $|\alpha|^2$ bzw.

$|\beta|^2$. Im ersten Fall erreicht das Qubit den Zustand $|0\rangle$ und im zweiten Fall den Zustand $|1\rangle$. Messungen hängen stets auch von der gewählten Basis für die Darstellung des Qubits ab. Bei der Messung von $|\varphi\rangle$ bezüglich der Dualbasis B' ergibt sich

$$|0\rangle = \frac{1}{\sqrt{2}}\left(|0'\rangle + |1'\rangle\right), \quad |1\rangle = \frac{1}{\sqrt{2}}\left(|0'\rangle - |1'\rangle\right)$$

$$|\psi\rangle = \frac{a}{\sqrt{2}}\left(|0'\rangle + |1'\rangle\right) + \frac{b}{\sqrt{2}}\left(|0'\rangle - |1'\rangle\right) = \frac{a+b}{\sqrt{2}}\,|0'\rangle + \frac{a-b}{\sqrt{2}}\,|1'\rangle$$

$|0'\rangle$ hat also die Wahrscheinlichkeit $\frac{(a+b)^2}{2}$ und $|1'\rangle$ die Wahrscheinlichkeit $\frac{(a-b)^2}{2}$. Wenn wir wissen, dass $|\psi\rangle = |0\rangle$ ist, d.h. $a = 1$ und $b = 0$, würden diese beiden Wahrscheinlichkeiten beide den Wert $\frac{1}{2}$ ergeben. Das bedeutet, dass die Messung von Qubit $|\varphi\rangle = |0\rangle$ ein probabilistisches Bit simuliert.

1.5 Operationen

Operationen an Qubits werden mathematisch dargestellt als Matrixmultiplikation mit unitären Matrizen U, für die gilt, dass $U^t\,U = I$. Für eine Matrix A ist A^t die Hermitesche Konjugierte von A, d.h. die transponierte Matrix mit konjugiert komplexen Elementen. A^t kann auch definiert werden als der Operator auf V mit der Eigenschaft, dass für alle $|v\rangle, |w\rangle \in V$ $(|v\rangle, A|w\rangle) = (A^t|v\rangle, |w\rangle)$ gilt.

Unitäre Matrizen haben den Vorteil, dass sie das innere Produkt zwischen Vektoren erhalten, es gilt z.B.

$$(U|\psi\rangle, U|\varphi\rangle) = (U^t\,U|\psi\rangle, |\varphi\rangle) = (|\psi\rangle, |\varphi\rangle)$$

Deshalb können mit Operationen Schaltkreise simuliert werden, z.B. die Hadamard-Transformation H oder die Pauli-Matrizen X, Y und Z:

$$H = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Theoretisch steht auch hier eine unendliche Menge an Gattern zur Verfügung, was auch noch einmal durch den folgenden Satz verdeutlicht wird.

Satz 1.5.1 *Jede 2-dimensionale unitäre Matrix U kann geschrieben werden als*

$$U = e^{i\gamma}\begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix}\begin{pmatrix} \cos\theta & i\,\sin\theta \\ i\,\sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} e^{i\beta} & 0 \\ 0 & e^{-i\beta} \end{pmatrix}.$$

1.6 2-Qubits

Ein 2-Qubit-*Register* wird durch das Tensorprodukt von zwei Qubits dargestellt. Das Tensorprodukt $x \otimes y$ von zwei Vektoren $x = (x_1 \dots x_m), y = (y_1 \dots y_n)$ ist der mn-Vektor $(x_1y_1, \dots, x_1y_m, x_2y_1, \dots, x_2y_n, \dots, x_my_i, \dots, x_my_n)$. Als Basis werden dann die Vektoren

$$|0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |1\rangle = |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |2\rangle = |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \text{ und}$$

$$|3\rangle = |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

verwendet. Damit gilt für zwei Vektoren $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ und $|\varphi\rangle = c\,|0\rangle + d\,|1\rangle$:

$$|\psi\rangle \otimes |\varphi\rangle = |\psi\rangle\,|\varphi\rangle = |\psi\ \varphi\rangle = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} = ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle$$

Das Tensorprodukt hat die folgenden Eigenschaften:

$$|v\rangle \otimes |w\rangle + |v\rangle \otimes |u\rangle = |v\rangle \otimes (|w\rangle + |u\rangle)$$

$$|v\rangle \otimes |w\rangle + |u\rangle \otimes |w\rangle = (|v\rangle + |u\rangle) \otimes |w\rangle$$

$$a(|v\rangle \otimes |w\rangle) = a|v\rangle \otimes |w\rangle = |v\rangle \otimes a|w\rangle$$

$$(|u\rangle \otimes |v\rangle, |w\rangle \otimes |x\rangle) = (|u\rangle, |w\rangle) \otimes (|v\rangle, |x\rangle)$$

Es gilt analog, dass $|00\rangle$ mit Wahrscheinlichkeit $|ac|^2$, $|01\rangle$ mit $|ad|^2$ usw. auftritt. Die Wahrscheinlichkeit, dass das erste Bit den Wert 0 annimmt, ist dementsprechend $|ac|^2 + |ad|^2$. In diesem Fall kann die verbleibende Wahrscheinlichkeit für das zweite Bit über den Post-Messung-Zustand

$$\frac{ac\,|00\rangle + ad\,|01\rangle}{\sqrt{|ac|^2 + |ad|^2}}$$

ausgedrückt werden, wobei der Nenner lediglich eine erneute Normierung darstellt.

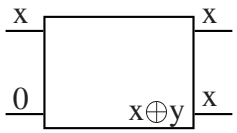
1.7 Transformationen

Transformationen sind gewissermaßen Operationen für 2-Qubits und werden durch 4x4-Matrizen dargestellt, wie z.B. folgende XOR-Transformation

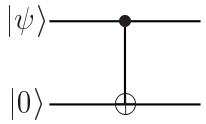
$$XOR = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Die schematische Darstellung erfolgt durch Quantenschaltkreise; ein solcher hat für die XOR-Transformation untenstehende Form. Zum Vergleich ist daneben ein funktionsgleicher „klassischer“ Schaltkreis aufgezeichnet.

Die XOR-Transformation führt also jeweils die 2-Qubits $|00\rangle \rightarrow |00\rangle$, $|01\rangle \rightarrow |01\rangle$, $|10\rangle \rightarrow |11\rangle$ und $|11\rangle \rightarrow |10\rangle$ ineinander über.
Mit „klassischen“ Schaltkreisen stellt es kein Problem dar, Bits zu kopieren, ein möglicher Schaltkreis wäre z.B.



Versucht man diesen Schaltkreis jedoch auf Qubits zu übertragen, so ist dies zum Scheitern verurteilt:



Wenn wir annehmen, dass $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$, so ergibt die Transformation rechnerisch

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ 0 \\ b \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ 0 \\ 0 \\ b \end{pmatrix} = a\,|00\rangle + b\,|11\rangle$$

Das 2-Qubit $|\psi\,\psi\rangle$ wäre jedoch $|\psi\,\psi\rangle = a^2\,|00\rangle + ab\,|01\rangle + ab\,|01\rangle + b^2\,|11\rangle$, d.h. zumindest mit dieser Transformation ist das Kopieren eines Qubits nicht möglich. Tatsächlich ist es überhaupt nicht möglich, ein Qubit zu kopieren, da es keine unitäre Matrix U gibt mit $U\,|\psi, 0\rangle = |\psi\,\psi\rangle$. Dies beweist der folgende

Satz 1.7.1 (No-cloning-Satz) *Es gibt keine Unitärtransformation U, die ein Qubit kopieren kann.*

Beweis: Sei U eine solche Transformation, so dass für Qubits $|\psi\rangle$ und $|\varphi\rangle$ gilt $U\,|\psi, 0\rangle = |\psi\,\psi\rangle$ und $U\,|\varphi, 0\rangle = |\varphi\,\varphi\rangle$. Dann folgt durch die Definition der unitären Matrizen und des Tensorprodukts

$$\langle\varphi|\psi\rangle\,\langle\varphi|\psi\rangle = \langle\varphi\varphi|\psi\psi\rangle = \langle U|\varphi 0\rangle\,|\,U|\psi 0\rangle\rangle = \langle U^t U|\varphi 0\rangle\,|\,\psi 0\rangle\rangle = \langle\varphi 0\,|\,\psi 0\rangle = \langle\varphi|\psi\rangle\,\langle 0|0\rangle$$

Da $\langle 0|0\rangle$ die Norm 1 hat, folgt aus dieser Umformung, dass $\langle\varphi|\psi\rangle^2 = \langle\varphi|\psi\rangle$ und somit gibt es solche Transformationen nur für die Fälle, in denen $\langle\varphi|\psi\rangle$ den Wert 0 oder 1 annimmt, was im allgemeinen nicht sinnvoll ist. □

1.8 n-Qubit-Register

n-Qubit-Register werden im 2^n-dimensionalen Raum betrachtet, etwa mit der Standardbasis

$$B = \{|i\rangle \mid i \in \{0,1\}^n\} = \{|i\rangle \mid 0 \leq i < 2^n\},$$

in dieser Notation gilt also beispielsweise $|101\rangle = |5\rangle$. Somit wird ein n-Qubit $|\psi\rangle$ notiert als

$$|\psi\rangle = \sum_{i\in\{0,1\}^n} \alpha_i\,|i\rangle \quad \text{, mit} \quad \sum_{i\in\{0,1\}^n} |\alpha_i|^2 = 1$$

Transformationen von n-Qubits sind dementsprechend Multiplikationen mit $2^n \times 2^n$-Matrizen. Dabei besteht eine Transformation der Form $M\,|\psi\rangle$ letztendlich aus $2^n(2\,2^n - 1)$ Operationen; ist U eine Transformation auf einem Qubit i, so gilt für die Anwendung dieser Transformation auf ein n-Qubit der Ausdruck

$$\bigotimes_{k=1}^{i-1} I \otimes U \otimes \bigotimes_{k=i+1}^{n} I.$$

Hierbei wird das Tensorprodukt für Matrizen verwendet, das für Matrizen A $(m \times n)$ und B $(p \times q)$ definiert ist als

$$A \otimes B = \begin{pmatrix} \overbrace{\begin{matrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \dots & \dots & \dots & \dots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{matrix}}^{nq} \end{pmatrix} mp.$$

Auch bei n-Qubits kann man die Dualbasis (auch Fourier-Basis genannt) verwenden, es gilt dann einfach

$$B' = \{|i'\rangle \mid 0 \leq i < 2^n\}.$$

Die Hadamard-Transformation für n-Qubits ist einfach $H_n = \bigotimes_{i=1}^{n} H$, also z.B. für $n = 2$

$$H_2 = H \otimes H = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

Für beliebige n kann auch folgende vereinfachte Darstellung verwendet werden, um etwa die Amplitude einer bestimmten Eingabe zu erhalten:

$$H_n |0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$$

bzw. wieder verallgemeinert auf allgemeine x

$$H_n |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{xy} |y\rangle \text{ , wobei } \quad xy = \bigoplus_{i=1}^{n} x_i \, y_i$$

Damit kann man beispielsweise für $x = |10110\rangle$ und $y = 11000$ mit $xy = 1$ (der Parität der bit-weisen UND-Verknüpfung von x und y) darauf schließen, dass die Amplitude von x ein negatives Vorzeichen hat und somit $-\frac{1}{\sqrt{2^5}}$ beträgt.

Kapitel 2

Quantenschaltkreise

2.1 Reversibilität

Ein wesentliches Merkmal von Quantenschaltkreisen ist, dass sie wegen der Verwendung von unitären Matrizen stets umkehrbar sind (für unitäre Matrizen gilt $U^t = U^{-1}$). Dies bedeutet, dass bei Kenntnis des Ergebnisses und der Transformation auf die Eingabe zurückgeschlossen werden kann, wie in folgendem Beispiel deutlich wird:

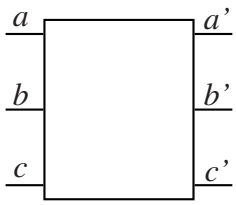
$$U \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} \rightarrow \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = U^t \begin{pmatrix} \alpha' \\ \beta' \end{pmatrix}$$

Dies stellt einen fundamentalen Unterschied zu klassischen Schaltkreisen dar, da z.B. bei einem einfachen UND-Gatter bei Ausgabe des Wertes 0 kein Rückschluss möglich ist, welcher Eingabewert 0 war. Gleichzeitig impliziert die Eigenschaft der Reversibilität auch, dass Eingänge und Ausgänge von Quantenschaltkreisen immer von derselben Dimension sein müssen.
Damit stellt sich die Frage, wie Quantenschaltkreise einzuordnen sind, also ob sie weniger mächtig, gleichmächtig oder mächtiger sind als bisher bekannte Schaltkreise. Tatsächlich ist es nicht schwierig einzusehen, dass Quantenschaltkreise ohne weiteres klassische und probabilistische Schaltkreise simulieren können, was im folgenden gezeigt werden soll.

2.2 Simulation klassischer Schaltkreise

Zur Simulation von klassischen Schaltkreisen wird üblicherweise ein spezielles Gatter verwendet:

Definition 2.2.1 (Fredkin-Gatter) *Ein Fredkin-Gatter ist ein Gatter der Form*



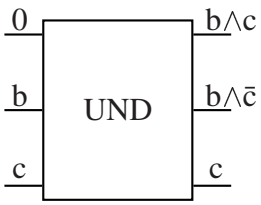
und den Regeln

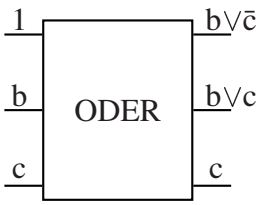
1. $f(a,b,0) = (a,b,0)$
2. $f(a,b,1) = (b,a,1)$

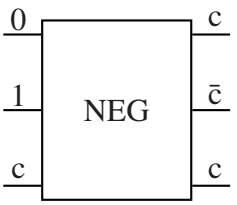
Als Folgerung aus dieser Definition ergibt sich, dass $f\,f = id$, wie man leicht durch Einsetzen nachvollziehen kann.
Die Matrix eines Fredkin-Gatters ist

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

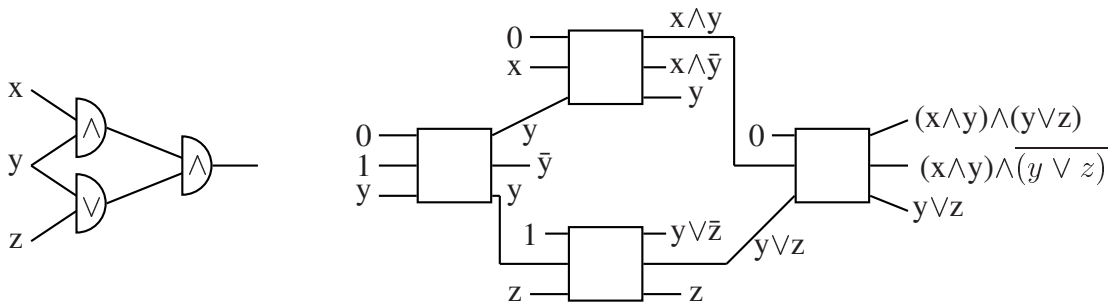
Mit Hilfe von Fredkin-Gattern können unter Verwendung von Konstanten (auch „ancillas“ genannt) die elementaren Funktionen UND, ODER und NEG simuliert werden:



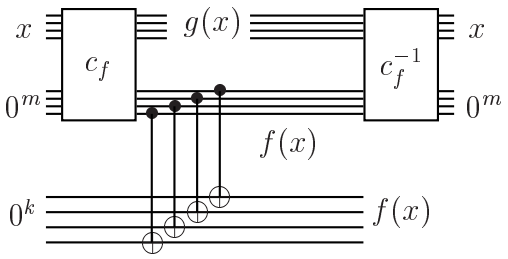




Mit diesen Gattern können nun auch komplexere Schaltkreise konstruiert werden, ein einfacheres Beispiel ist folgender klassischer Schaltkreis und die entsprechende Quantenschaltkreis-Simulation:



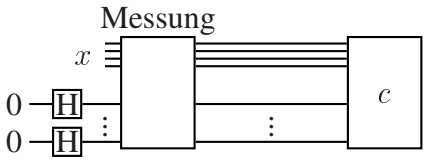
Wenn wir uns einen solchen Quantenschaltkreis als „Blackbox“ vorstellen, so haben wir auf der Eingabeseite neben einem Eingabevektor x noch eine Reihe von Konstanten 0 und 1, auf der Ausgabeseite zum einen die gewünschte Funktion $f(x)$, zum anderen aber auch einen nicht unbedeutenden Teil an ungenutzten Ausgängen, die hier mit $g(x)$ bezeichnet werden sollen („garbage“). Die Einsen in der Eingabe können durch Nullen und Negationsgatter (die auch reversibel sind) ersetzt werden. Es ist aber unschön (und kann in Quantenschaltkreisen zu Problemen führen), dass die zusätzlichen Ausgänge auch von der Eingabe x abhängen; dies kann allerdings vermieden werden, indem man eine etwas aufwändigere Schaltung konstruiert:



Auf diese Weise erhält man durch Hinzufügen einer bekannten Anzahl weiterer Konstanten 0 außer der Eingabe selbst nur die Funktion $f(x)$ zurück, d.h. es gilt $(x, 0^m, 0^k) \rightarrow (x, 0^m, f(x))$ bzw. unter Vernachlässigung der Konstanten $x \rightarrow (x, f(x))$. Dies ließe sich unter Verwendung der obigen CNOT-Verknüpfung auch mit einer weiteren k-dimensionalen Variablen y anstatt den Konstanten 0^k verbinden, so dass man die Wirkung $(x, y) \rightarrow (x, y \oplus f(x))$ erhalten würde (mit bit-weiser Parität).

2.3 Simulation probabilistischer Schaltkreise

Damit ist auf jeden Fall klar, dass Quantenschaltkreise klassische Schaltkreise simulieren können. Wie sieht das ganze nun jedoch bei probabilistischen Schaltkreisen aus? Ein probabilistischer Schaltkreis lässt sich schematisch so darstellen, dass man neben fest gegebenen Eingabewerten eine Reihe zufällig gewählter Konstanten verwendet und der Schaltkreis dann mit einer abschätzbar großen Wahrscheinlichkeit das korrekte Ergebnis liefert. Es ist allerdings ziemlich einfach, mit Quantenschaltkreisen zufällig gewählte Konstanten ins Spiel zu bringen, etwa mit folgendem Schaltkreis:



Bevor die eigentliche Funktion ausgeführt wird, werden die Konstanten 0 jeweils einzeln Hadamard-transformiert, bei n Konstanten gilt dann

$$|0^n\rangle \xrightarrow{H} \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle.$$

Durch die Messung erhält man einen Konstanten-Vektor $r \in \{0,1\}^n$, der sich zufällig unter Gleichverteilung ergibt, da sich die Wahrscheinlichkeit für einen konkreten Zustand $|i\rangle$ wie bereits gesehen zu $|\frac{1}{\sqrt{2^n}}|^2 = \frac{1}{2^n}$ ergibt.

2.4 Die Komplexitätsklasse BQP

In diesem Kapitel geht es darum, einen Nachweis für die Einordnung der Komplexitätsklasse BQP (*bounded error quantum polynomial time*) zu erbringen. Zunächst müssen wir diese Klasse jedoch formal definieren.

Definition 2.4.1 (BQP) *Sei $L \subseteq \Sigma^*$. Dann gilt $L \in BQP$ genau dann, wenn es eine „uniforme“ Familie von Quantenschaltkreisen $\{c_1, c_2, \ldots\}$ und ein Polynom p gibt, so dass $\forall\, x \in \Sigma^*$ gilt:*

- $x \in L \Rightarrow Prob($*Ausgabe von* $c_{|x|}$ *mit Eingabe* x *ist* $1) \geq \frac{3}{4}$
- $x \notin L \Rightarrow Prob($*Ausgabe von* $c_{|x|}$ *mit Eingabe* x *ist* $0) \geq \frac{3}{4}$

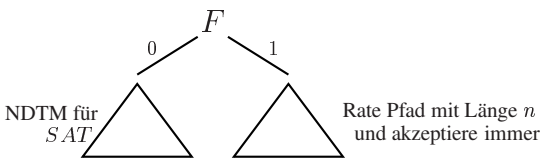
- c_n *hat höchstens* $p(n)$ *Gatter.*

Von den Resultaten in 2.2 und 2.3 folgt, dass $P \subseteq BPP \subseteq BQP$ (die Klasse BPP = *bounded error probabilistic polynomial time* ist die Klasse der polynomiell beschränkten probabilistischen Algorithmen mit durch eine Konstante beschränktem Fehler). Was wir an dieser Stelle zeigen wollen, ist die Beziehung $BQP \subseteq PP$, wobei PP (*probabilistic polynomial time*) folgende Definition hat:

Definition 2.4.2 *Sei* $L \subseteq \Sigma^*$. *Dann gilt* $L \in PP$ *genau dann, wenn es eine polynomiell beschränkte nichtdeterministische Turingmaschine* M *gibt mit den Eigenschaften*

- $x \in L \Rightarrow$ *Anzahl der akzeptierenden Pfade in* $M(x) >$ *Anzahl der verwerfenden Pfade in* $M(x)$
- $x \notin L \Rightarrow$ *Anzahl der akzeptierenden Pfade in* $M(x) \leq$ *Anzahl der verwerfenden Pfade in* $M(x)$

Als Beispiel kann hier wieder einmal das Erfüllbarkeitsproblem dienen: es gilt $SAT \in PP$, da man eine recht einfache Maschine M konstruieren kann:



Die nichtdeterministische Turingmaschine für das SAT-Problem akzeptiert bekanntlich jede Eingabe, die erfüllbar ist. Somit gilt für die obige Maschine M:

- $F \in SAT \Rightarrow$ Anzahl akzeptierender Pfade $\geq 2^n + 1$ (alle Pfade im rechten Dreieck und mindestens einer im linken)
- $F \notin SAT \Rightarrow$ Anzahl akzeptierender Pfade $= 2^n =$ Anzahl verwerfender Pfade (die NDTM für SAT findet keine erfüllende Belegung)

Somit genügt diese Maschine den Kriterien aus PP; da es zudem sehr einfach ist, die Pfade zu zählen, gehört das Problem auch zur Klasse $PSPACE$, der Klasse der Probleme mit polynomiellem Speicherbedarf.
Wir definieren nun noch zwei weitere Klassen bzw. Mengen, die wir dann in mehreren Lemmata verwenden.

Definition 2.4.3 (FP) *FP bezeichnet die Menge der polynomiell berechenbaren Funktionen:*

$$FP = \{f : \Sigma^* \to \mathbb{N} \mid f \text{ ist in polynomieller Zeit berechenbar}\}.$$

Definition 2.4.4 ($\#P$) *Die Funktionenklasse $\#P$ ist definiert über*

$$\#P = \{f : \Sigma^* \to \mathbb{N} \mid \text{ es gibt eine polynomiell beschränkte NDTM } M, \text{ so dass } \forall\, x\ f(x) = \text{Anzahl der akzeptierenden Pfade von } M(x)\}.$$

Ein Beispiel für eine Funktion $f \in \#P$ ist:

$$f(x) = \begin{cases} \text{Anzahl erfüllende Belegungen von } F & \text{falls } x \text{ eine Formel } F \text{ kodiert} \\ 0 & \text{sonst} \end{cases}$$

Lemma 2.4.5 *Sei $f \in FP$. Dann gilt stets auch $f \in \#P$.*

Wir wollen hier nur kurz den Beweis andeuten: Wenn gilt $f(x) \in FP$, so muss $f(x) \leq 2^{p(|x|)}$ sein für ein Polynom p. Wir konstruieren nun eine nichtdeterministische Turingmaschine:

- Eingabe x
- Berechnung von $f(x)$
- Raten einer Zahl $i \in \{1, \ldots, 2^{p(|x|)}\}$
- Akzeptieren $\Leftrightarrow i \leq f(x)$

Mit dieser Maschine ist gewährleistet, dass $f(x)$ für jede Eingabe x die Anzahl der akzeptierenden Pfade repräsentiert und somit gilt $f(x) \in \#P$. □

Lemma 2.4.6 *Sei $L \in PP$. Dann gibt es ein $f \in \#P$ und ein $g \in FP$ mit der Eigenschaft $\forall\, x :\ x \in L \Leftrightarrow f(x) > g(x)$.*

Lemma 2.4.7 *Seien $f_1,\ f_2 \in \#P$. Dann sind auch $f_1 + f_2$ und $f_1 \cdot f_2 \in \#P$. Ferner gilt für $f_1(x) =$ Anzahl akzeptierender Pfade von $M(x)$ mit Laufzeit(M) $\leq p(|x|)$, dass $2^{p(|x|)} - f_1(x) \in \#P$, was anschaulich ein Umdrehen der akzeptierenden Blätter in der Baumdarstellung von M ist.*

Wir betrachten nun eine Menge $U = \{H, R_{\frac{\pi}{2}}, CNOT, Toffoli\}$, die eine universelle Menge von Quanten-Gattern darstellt. Zur Erinnerung:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, R_{\frac{\pi}{2}} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

Das Controlled-Not-Gatter funktioniert auf zwei Bits, das Toffoli-Gatter auf drei Bits. Die Gatter haben die Gestalt

und ihre analytischen Definitionen sind

$$CNOT(a,b) = (a, a \oplus b) \text{ und } Toffoli(a,b,c) = (a,b,c \oplus (a \wedge b)).$$

Satz 2.4.8 *U ist eine universelle Menge von Quanten-Gattern. D.h. für $L \in BQP$ mit einer Familie von Quantenschaltkreisen $\{c_1, c_2, \ldots\}$, die L mit konstantem Fehler ϵ akzeptiert, gibt es eine Schaltkreisfamilie $\{c'_1, c'_2, \ldots\}$, bei der für jedes n c'_n nur aus Gattern aus U besteht. Sie hat ebenfalls die Fehlerwahrscheinlichkeit ϵ und ist in der Größe durch $p(|c_n|, \log(\frac{1}{\epsilon}))$ beschränkt, d.h. polynomiell in Abhängigkeit von c_n und ϵ.*

Nach diesen Vorüberlegungen können wir nun endlich zu unserem eigentlichen Anliegen, dem Beweis $BQP \subseteq PP$, kommen:
Sei $L \in BQP$ und $\{c_1, c_2, \ldots\}$ eine Familie von Quantenschaltkreisen für L. $h(1^n)$ bezeichne die Anzahl von H-Gattern in c_n; es gilt $h \in FP$. Wir beschreiben nun den Quantenschaltkreis als Überlagerung von Zuständen $\sum_s \alpha_s^t |s\rangle$ mit $s \in \{0,1\}^m$ und α_s^t die Amplitude von $|s\rangle$ nach t Schritten. Der Anfangszustand sei $|x0\ldots1\ldots\rangle = |s_i^0\rangle$, der akzeptierende Zustand $|s_a^t\rangle = |x10\ldots1\rangle$.

Lemma 2.4.9 *Es gibt Funktionen R_+, R_-, I_+ und I_- in $\#P$ mit der Eigenschaft $\forall x \in \{0,1\}^n \; \forall s \in \{0,1\}^m \; \forall t : \; \alpha_s^t = \frac{1}{\sqrt{2}^{h(1^n)}} (R_+(x,s,t) - R_-(x,s,t) + i\, I_+(x,s,t) - i\, I_-(x,s,t))$.*

Mit unserer universellen Menge U können wir den Quantenschaltkreis darstellen als $c_{|x|}(x) = U_{t-1} \ldots U_2\, U_1\, U_0\, |s_i^0\rangle$, wobei $U_i \in \{H, R, CNOT, Toffoli\}$. Wir können jetzt einen Graphen konstruieren, der als Knoten die möglichen Konfigurationen des Schaltkreises hat, also die Paare (i,s) mit $1 \leq i \leq t$ und $s \in \{0,1\}^m$. Es gibt eine Kante von (i,s) nach $(i+1,s')$ falls die Transformation U_i Zustand $|s\rangle$ in Zustand $|s'\rangle$ transformieren kann. Die Kante hat ein Gewicht $a_{s'}$ für die Wahrscheinlichkeit, dass $|s\rangle$ in $|s'\rangle$ transformiert wird. Formal gilt also $U_i |s\rangle = \sum_{s'} a_{s'} |s'\rangle$.

Analysieren wir, welche Ausgangskanten für die möglichen Knoten in Frage kommen, so erhält man für das $CNOT$- und das $Toffoli$-Gatter jeweils eine Kante mit Wert 1, für $R_{\frac{\pi}{2}}$ eine Kante mit Wert i oder 1. Lediglich die Hadamard-Transformation ist etwas komplizierter: sie spaltet den Zustand in zwei Teile mit Amplitude $\frac{1}{\sqrt{2}}$ auf, wobei in einem Fall auch ein negatives Vorzeichen ins Spiel kommt.
Somit besitzt H 2 Kanten mit $\pm\frac{1}{\sqrt{2}}$, es liegt also immer nur bei den Hadamard-Gattern eine Verzweigung vor und ein Pfad von $|s_i^0\rangle$ nach $|s_j^k\rangle$ im Schaltkreis kann mit genauso vielen Bits definiert werden, wie H-Gatter bis zum Schritt k vorkommen. Die Amplitude von $|s_j^k\rangle$ ist dann

$$\alpha_{s_j}^k = \sum_y \frac{1}{\sqrt{2}^{h(k)}}\zeta_y,$$

wobei die y die Pfade von $|s_i^0\rangle$ nach $|s_j^k\rangle$ symbolisieren und ζ_y ein Produkt der Werte in y darstellt, also $\zeta_y \in \{1, -1, i, -i\}$.
Dasselbe gilt dann auch für die Amplitude des akzeptierenden Zustands $|s_a^t\rangle$:

$$\alpha_a^t = \sum_y \frac{1}{\sqrt{2}^{h(k)}}\zeta_y.$$

Andererseits gilt, dass

$$\alpha_a^t = \frac{1}{\sqrt{2}^{h(1^n)}}\,(R_+ - R_- + i\,I_+ - i\,I_-),$$

wobei R_+ die Anzahl der Pfade y mit Produkt 1, R_- die Anzahl der Pfade y mit Produkt -1, I_+ mit Produkt i und I_- mit Produkt $-i$ darstellen; außerdem sind wegen des letzten Lemmas, alle Funktionen $\in \#P$.
Damit können wir die Wahrscheinlichkeit für den akzeptierenden Zustand ermitteln:

$$Prob(c(x) = 1) = |\alpha_a^t|^2 = \frac{1}{2^{h(1^n)}} \underbrace{(R_+^2 - R_-^2 + 2\,R_+R_- + I_+^2 + I_-^2 - 2\,I_+I_-)}_{g_1(x)-g_2(x)}$$

Wegen Lemma 2.4.7 gilt g_1, $g_2 \in \#P$. Da laut Voraussetzung $L \in BQP$ gilt

- $x \in L \Rightarrow \frac{g_1(x)-g_2(x)}{2^{h(1^n)}} \geq \frac{3}{4} \Rightarrow g_1(x) - g_2(x) \geq \frac{3}{4}\,2^{h(1^n)}$
- $x \notin L \Rightarrow \frac{g_1(x)-g_2(x)}{2^{h(1^n)}} \leq \frac{1}{4} \Rightarrow g_1(x) - g_2(x) \leq \frac{1}{4}\,2^{h(1^n)}$

Mit der Funktion $f(x) = 2^{q(|x|)} + g_1(x) - g_2(x)$, die wegen Lemma 2.4.7 ebenfalls $\in \#P$ ist, gilt schließlich

- $x \in L \Rightarrow f(x) \geq \frac{3}{4}\,2^{h(1^n)}$
- $x \notin L \Rightarrow f(x) \leq \frac{1}{4}\,2^{h(1^n)}$

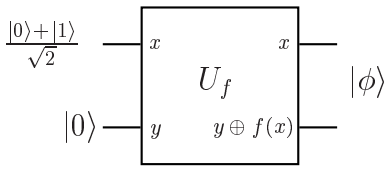
Wegen Lemma 2.4.6 folgt, dass $L \in PP$. □

Kapitel 3

Erste Quantenalgorithmen

3.1 Quanten-Parallelismus

Ein großer Vorteil von Quantenschaltkreisen ist, dass sie durch den Überlagerungszustand von Qubits die Durchführung von Transformationen parallel auf alle möglichen Eingaben ermöglicht. Dies wird am besten an einem Beispiel deutlich. Hier wird angenommen, dass wir eine "black box" U_f haben, die einen Schaltkreis für f simuliert. Wir wollen die Anzahl von Fragen, die an U_f gestellt werden müssen, berechnen.



Der Schaltkreis U_f stellt wie zuvor angedeutet eine Transformation $(x, y) \to (x, y \oplus f(x))$ dar. Der Ausdruck $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ stammt von einem Hadamard-transformierten Qubit $|0\rangle$. Insgesamt ergibt sich also

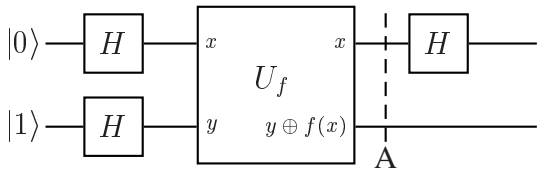
$$\frac{|0\rangle+|1\rangle}{\sqrt{2}}\,|0\rangle \;\to\; |\phi\rangle = \frac{|0\,f(0)\rangle+|1\,f(1)\rangle}{\sqrt{2}}.$$

Damit stecken alle möglichen Ergebnisse der Funktion in dem resultierenden Qubit $|\phi\rangle$. Dies lässt sich übertragen auf Funktionen $f : \{0,1\}^n \to \{0,1\}$, indem pro Eingabebit eine Hadamard-transformierte 0 verwendet wird. Als Ergebnis erhält man dementsprechend

$$|\phi\rangle = \sum_{i \in B^n} \frac{|i\,f(i)\rangle}{\sqrt{2^n}}.$$

3.2 Der Deutsch-Algorithmus

Als Vorüberlegung betrachten wir die Funktionsweise von folgendem Schaltkreis:



Die beiden Hadamard-Transformationen auf der linken Seite führen zunächst zu $|01\rangle \rightarrow \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$. Für die möglichen Ausgänge der Funktion f sind die Übergänge jeweils

$f(0)$	$f(1)$	Position A	Endposition
0	0	$(\frac{\|0\rangle+\|1\rangle}{\sqrt{2}})\,(\frac{\|0\rangle-\|1\rangle}{\sqrt{2}})$	$\|0\rangle(\frac{\|0\rangle-\|1\rangle}{\sqrt{2}})$
0	1	$(\frac{\|0\rangle-\|1\rangle}{\sqrt{2}})\,(\frac{\|0\rangle-\|1\rangle}{\sqrt{2}})$	$-\|1\rangle(\frac{\|0\rangle-\|1\rangle}{\sqrt{2}})$
1	0	$(\frac{-\|0\rangle+\|1\rangle}{\sqrt{2}})\,(\frac{\|0\rangle-\|1\rangle}{\sqrt{2}})$	$\|1\rangle(\frac{\|0\rangle-\|1\rangle}{\sqrt{2}})$
1	1	$(\frac{-\|0\rangle-\|1\rangle}{\sqrt{2}})\,(\frac{\|0\rangle-\|1\rangle}{\sqrt{2}})$	$-\|0\rangle(\frac{\|0\rangle-\|1\rangle}{\sqrt{2}})$

Wenn wir nun wissen, dass gilt $f(0) = f(1)$, d.h. dass $f(0) \oplus f(1) = 0$, können wir dies mit Hilfe der Ergebnisse aus der Überlegung auch darstellen als

$$(-1)^{f(1)}\,|f(0) \oplus f(1)\rangle\,(\tfrac{|0\rangle-|1\rangle}{\sqrt{2}})$$

Damit erhalten wir bei einer Messung von Bit 1 mit Wahrscheinlichkeit 1 den Wert von $f(0) \oplus f(1)$. Diese Vorüberlegung ist nützlich für das nachfolgende Problem.

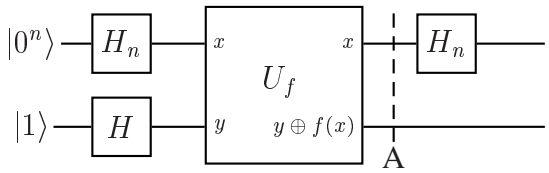
3.3 Deutsch-Josza-Problem

Die Problemstellung ist, dass für eine Funktion $f : \{0,1\}^n \rightarrow \{0,1\}$ bekannt ist, dass sie entweder (i) konstant ist oder (ii) ihre Wertemenge gleichviele 0en und 1en enthält (formal: $\|\{x \in \{0,1\}^n | f(x) = 0\}\| = \|\{x \in \{0,1\}^n | f(x) = 1\}\| = 2^{n-1}$). Weil wir wissen, dass eine der beiden Möglichkeiten zutrifft, spricht man auch von

einem Promise-Problem. Wir haben einen Schaltkreis, der f berechnet. Die Frage ist nun, wie effizient man die Frage, zu welchem Typ eine gegebene Funktion f gehört, beantworten kann. Wir wollen die Anzahl von Eingaben x bestimmen, für die $f(x)$ berechnet werden muss, um den Typ von f zu finden.
Deterministisch: Im schlimmsten Fall müssen hier $2^{n-1}+1$ Versuche von verschiedenen Eingabevektoren durchgeführt werden, um den Typ sicher zu bestimmen.
Probabilistisch: Bei probabilistischen Algorithmen muss man die Anzahl an Stichproben ermitteln, die man nehmen muss, um eine genügend kleine Fehlerwahrscheinlichkeit zu erhalten. Angenommen, f ist vom Typ (ii), aber für alle x, die sich aus der Stichprobe ergeben, ist $f(x)$ gleich. Die Wahrscheinlichkeit, nach $k+1$ Proben nur gleiche Funktionswerte zu erhalten, ist dann

$$P(\text{Fehler}) = 1\,(\tfrac{2^{n-1}-1}{2^n-1})(\tfrac{2^{n-1}-2}{2^n-2})...(\tfrac{2^{n-1}-k}{2^n-k}) < (\tfrac{1}{2})^k = \tfrac{1}{2^k}$$

Damit ist dieses Problem probabilistisch mit einer konstanten Anzahl von Stichproben an f lösbar.
Quantenschaltkreise: Die Lösung des Problems mit einem Quantenschaltkreis verwendet einen ganz ähnlichen Schaltkreis wie im vorigen Abschnitt:



Hier finden nun folgende Übergänge statt:

$$\begin{aligned}
|0^n\,1\rangle \quad &\rightarrow \quad \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\
&\rightarrow \quad \frac{1}{\sqrt{2^n}} \sum_{x} |x\rangle \,(\frac{|f(x)\rangle - |\overline{f(x)}\rangle}{\sqrt{2}}) \qquad (3.1) \\
&= \quad \frac{1}{\sqrt{2^n}} \sum_{x} |x\rangle \,(-1)^{f(x)}\,(\frac{|0\rangle - |1\rangle}{\sqrt{2}}) \\
&\overset{nachA}{\rightarrow} \quad \frac{1}{\sqrt{2^n}}\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} \sum_{z\in\{0,1\}^n} |z\rangle \,(-1)^{xz}\,(-1)^{f(x)}\,(\frac{|0\rangle - |1\rangle}{\sqrt{2}}) \\
&= \quad \frac{1}{2^n} \sum_{x,z\in\{0,1\}^n} |z\rangle \,(-1)^{xz+f(x)}\,(\frac{|0\rangle - |1\rangle}{\sqrt{2}})
\end{aligned}$$

In (3.1) ist $f(x)$ entweder 0 oder 1, d.h. $0 \rightarrow \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ oder $1 \rightarrow \frac{|1\rangle-|0\rangle}{\sqrt{2}} = \frac{-(|0\rangle-|1\rangle)}{\sqrt{2}}$.
Der letzte Ausdruck lässt sich vereinfacht schreiben als

$$\sum_z |z\rangle\, \alpha_z\, (\tfrac{|0\rangle - |1\rangle}{\sqrt{2}}) \quad \text{mit} \quad \alpha_z = \tfrac{1}{2^n} \sum_x (-1)^{xz+f(x)}$$

Für $|z\rangle = |0^n\rangle$ gilt $xz = 0$, weshalb die Amplitude zu $\alpha_z = \frac{1}{2^n} \sum_x (-1)^{f(x)}$ wird. Betrachten wir nun den Fall (i), also eine konstante Funktion f. Dann gilt $\alpha_z = \frac{1}{2^n}(\pm 2^n) = \pm 1$. Für den Fall (ii) hingegen heben sich die Vorzeichen in der Summe gerade auf und es gilt $\alpha_z = 0$. Bei einer Messung der ersten n Bits von $\sum_z |z\rangle\, \alpha_z\, (\frac{|0\rangle - |1\rangle}{\sqrt{2}})$ tritt nun jedes mögliche z mit Wahrscheinlichkeit $|\alpha_z|^2$ auf. Handelt es sich bei dem gemessenen Wert um 0^n, dann ist f vom Typ (i), da die Wahrscheinlichkeit hierfür 1 ist. Gilt hingegen $z \neq 0^n$, so ist f vom Typ (ii).
Mit dem Quantenschaltkreis kann dieses Problem damit parallel mit nur einer Stichprobe gelöst werden.

3.4 Simon's Algorithmus

Problemstellung: Sei $f : \{0,1\}^n \to \{0,1\}^n$ eine Funktion, für die es ein $a \in \{0,1\}^n$ (das „Geheimwort") gibt mit

1. $\forall x :\ f(x) = f(x \oplus a)$, wobei $a \neq 0^n$
2. $\forall x \forall y\ x \neq y$ und $x \neq y \oplus a \Rightarrow f(x) \neq f(y)$

(Man spricht in diesem Zusammenhang auch von einer 2-zu-1-Funktion, da jeweils genau 2 Elemente unter der Funktion dasselbe Bild haben.)
Wir haben eine "black box" U_f für f und die Frage ist nun wieder, wie komplex ein Algorithmus wird, der dieses a für eine gegebene Funktion f herausfindet. Wie wir sehen werden, kann dies deterministisch und probabilistisch bestenfalls mit $\Omega(2^{\frac{n}{2}})$ geschehen, mit einem Quantencomputer hingegen in einer Laufzeit $O(n)$.

Deterministisch: Eine deterministische Vorgehensweise müsste zwangsläufig systematisch mögliche Eingaben für f durchprobieren, bis das Ergebnis zweier Eingaben gleich ist. Wenn z.B. gleich zu Beginn $f(x_1) = f(x_2)$ wäre, so müsste gelten $x_2 = x_1 \oplus a$ und damit $a = x_1 \oplus x_2$. Für drei Elemente müsste dann für alle Kombinationen überprüft werden, ob zwei Funktionswerte gleich sind. Ist dies nicht der Fall, so sind bereits 3 mögliche Werte für a eliminiert. Allgemein gilt: wenn für unterschiedliche $x_1, ..., x_k$ auch alle Funktionswerte $f(x_1), ..., f(x_k)$ paarweise unterschiedlich sind, können $\binom{k}{2}$ mögliche Werte für a eliminiert werden. Damit nur noch ein mögliches a übrigbleibt, muss für k also gelten $2^n - 1 - \binom{k}{2} \stackrel{!}{=} 1$ und somit $k = \Omega(2^{\frac{n}{2}})$.

Probabilistisch: Bei probabilistischer Vorgehensweise ist es sinnvoll, ein

Element a zufällig unter Gleichverteilung aus $\{0,1\}^n - \{0\}^n$ zu wählen. Wenn wir nun wieder k Werte $x_1, ..., x_k$ betrachten, für die $f(x_1), ..., f(x_k)$ paarweise unterschiedlich sind und mit a' ein Element bezeichnen, das noch nicht eliminiert wurde, d.h. $a' \neq x_1 \oplus x_2, a' \neq x_1 \oplus x_3, ..., a' \neq x_{k-1} \oplus x_k$, dann gilt unter Verwendung der Formel von Bayes

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

für die Wahrscheinlichkeit, dass es sich bei a' um das gesuchte Element a handelt

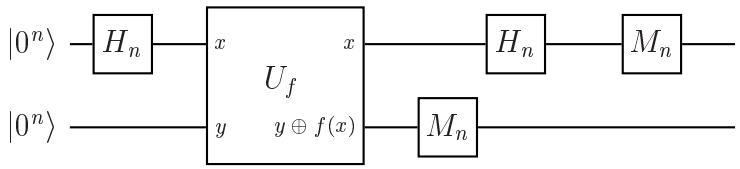
$$P(a = a' \mid f(x_1), ..., f(x_k) \text{ alle ungleich}) = \frac{P(f(x_1),...,f(x_k) alle\ ungleich | a=a')\, P(a=a')}{P(f(x_1),...,f(x_k) alle\ ungleich)}$$

Die einzelnen Wahrscheinlichkeiten in diesem Ausdruck bzw. ihre Abhängigkeit von a' erhält man durch einfache Überlegungen:

- $P(f(x_1), ..., f(x_k) \text{ alle ungleich} \mid a = a') = 1$, da a' noch nicht eliminiert wurde
- $P(a = a')$ ist gleich für alle Elemente a', die noch nicht eliminiert wurden
- $P(f(x_1), ..., f(x_k) \text{ alle ungleich})$ ist unabhängig von a'

Damit ist klar, dass $P(a = a' \mid f(x_1), ..., f(x_k) \text{ alle ungleich})$ für alle Elemente a', die noch nicht eliminiert wurden, gleich ist. Somit müssen wir ebenfalls $\Omega(2^{\frac{n}{2}})$ Stichproben machen, um ein genügend wahrscheinliches Resultat zu erhalten.

Lösung von Simon: Der Quantenschaltkreis zur Lösung des Problems ist nachfolgend abgebildet:



M_n bedeutet, dass an dieser Stelle eine Messung erfolgt. Es finden zunächst folgende Transformationen statt:

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

Die Messung der letzten n Qubits ergibt den Wert für $f(x)$, von dem wir allerdings wissen, dass mit dem gesuchten a gilt $f(x) = f(a \oplus x)$. Deshalb ergibt sich für einen bekannten Funktionswert nach Normierung als Post-Messung-Zustand

$$\frac{|x\rangle+|x\oplus a\rangle}{\sqrt{2}}\,|f(x)\rangle$$

Die Anwendung von H_n auf die ersten n Qubits ergibt

$$\begin{aligned}
\frac{1}{\sqrt{2}\sqrt{2^n}}\left[\sum_{z\in\{0,1\}^n}|z\rangle\,(-1)^{xz}+\sum_{z\in\{0,1\}^n}|z\rangle\,(-1)^{(x\oplus a)z}\right]|f(x)\rangle &= \\
\frac{1}{\sqrt{2^{n+1}}}\left[\sum_{z\in\{0,1\}^n}|z\rangle\,((-1)^{xz}+(-1)^{(x\oplus a)z})\right]|f(x)\rangle &= \\
\frac{1}{\sqrt{2^{n+1}}}\left[\sum_{z\in\{0,1\}^n}|z\rangle\,(-1)^{xz}(1+(-1)^{az})\right]|f(x)\rangle &= \\
\sum_{z\in\{0,1\}^n}\alpha_z\,|z\rangle \text{ mit } \alpha_z=\frac{(-1)^{xz}(1+(-1)^{az})}{\sqrt{2^{n+1}}}&
\end{aligned}$$

Falls nun $za=0$, so gilt $\alpha_z=\frac{(-1)^{xz}}{\sqrt{2^{n-1}}}=\pm\frac{1}{\sqrt{2^{n-1}}}$, ist hingegen $za=1$, so ist $\alpha_z=0$. Dies bedeutet, dass wir im Post-Messung-Zustand nach der zweiten Messung auf jeden Fall ein z erhalten, für das $za=0$ gilt, da nur diese z mit echt positiver, gleicher Wahrscheinlichkeit verteilt sind. Es gilt für die Komponenten von jedem dieser z jeweils, dass

$$z_1a_1\oplus z_2a_2\oplus\ldots\oplus z_n\,a_n=0.$$

Dies führt letztendlich zu einem linearen Gleichungssystem, dessen Zeilen alle die obige Form haben. Die z's gehören alle zu dem Teilraum von Vektoren in $\{0,1\}^n$, die zu a orthogonal sind. Dieser Teilraum hat Dimension $n-1$. Für $n-1$ linear unabhängige z hat das Gleichungssystem zwei Lösungen, 0 und a.
Wie bekommt man nun diese linear unabhängigen z? Stellen wir uns k linear unabhängige $z^{(1)}, z^{(2)}, ..., z^{(k)}$ vor. Dann gibt es höchstens 2^k Vektoren y, die linear abhängig von $\{z^{(1)}\ldots z^{(k)}\}$ sind (bei binären Vektoren ergibt sich dies aus der Potenzmenge). Wir betrachten nun die Wahrscheinlichkeit dafür, dass eine gleichverteilte Auswahl an z linear unabhängig ist:

- Am Anfang wählen wir $z^{(1)}$ mit Wahrscheinlichkeit $P=(1-\frac{1}{2^{n-1}})$, da $z=0^n$ nicht zulässig bzw. auf jeden Fall von jedem z linear abhängig ist, und es 2^{n-1} orthogonale Vektoren zu a gibt.

- dann $z^{(2)}$ mit Wahrscheinlichkeit $P=(1-\frac{2}{2^{n-1}})$, da hier schon 2 z in Frage kommen, die linear abhängig vom ersten sind (erneut 0^n und $z^{(1)}$ selbst)

- insgesamt ergibt sich mit unserer Vorüberlegung

$$\begin{aligned} P &= (1-\frac{1}{2^{n-1}})\,(1-\frac{2}{2^{n-1}})\,(1-\frac{4}{2^{n-1}})...(1-\frac{2^{n-2}}{2^{n-1}}) \geq \\ & \left[1-(\frac{1}{2^{n-1}}+\frac{1}{2^{n-2}}+...+\frac{1}{4})\right]\,\frac{1}{2} \geq \\ & (1-\frac{1}{2})\,\frac{1}{2} = \frac{1}{4} \end{aligned}$$

Die Wahrscheinlichkeit, dass nach n Iterationen die n Messungen linear unabhängig sind, ist also größer als $\frac{1}{4}$, bzw. die komplementäre Wahrscheinlichkeit für die lineare Abhängigkeit kleiner als $\frac{3}{4}$. Wenn wir das ganze nun k-mal wiederholen, so wird die Wahrscheinlichkeit für die lineare Abhängigkeit mit jedem weiteren Durchgang relativ schnell klein:

$$P(\text{k-mal sind die } n \text{ gemessenen Vektoren linear abhängig}) \leq (\tfrac{3}{4})^k < \tfrac{1}{2^{k-2}}$$

Wir können also in $k \cdot n$ Schritten garantieren, dass mit einer Wahrscheinlichkeit größer als $1-\frac{1}{2^{k-2}}$ das gesuchte a gefunden wird. Somit liegt die Laufzeit in $O(n)$.

Kapitel 4

Suchalgorithmus von Grover

4.1 Die Problematik

Das Problem, das der Suchalgorithmus von Grover löst, wird oft auch als „Suche nach einer Nadel im Heuhaufen“ bezeichnet. Digitalisiert kann man sich dies so vorstellen, dass man ein Array von n Bits vorliegen hat und darin nun einen einzigen Eintrag sucht, der nachprüfbare Bedingungen erfüllt.
Formal kann dies folgendermaßen ausgedrückt werden: Sei $f : \{0,1\}^n \rightarrow \{0,1\}$ eine Funktion, von der wir wissen, dass es genau ein a gibt mit $f(a) = 1$, d.h. $f(x) = 0 \ \forall x \neq a$. Für die Funktion liege nur eine „Blackbox“ in Form eines Schaltkreises U_f vor, der bei Anlegen eines Vektors x den Funktionswert $f(x)$ zurückliefert. Die Frage ist nun, wie viele Werte von x getestet werden müssen, um den Wert a zu finden, bei dem die Funktion 1 wird. Zur Vereinfachung der Schreibweise soll in den folgenden Argumentationen anstatt der Zahl 2^n stets die Bezeichnung N verwendet werden.

Deterministisch: Mit einem deterministischen Ansatz müsste im worst case $N - 1$ mal angefragt werden, um den Wert für a zu erhalten.

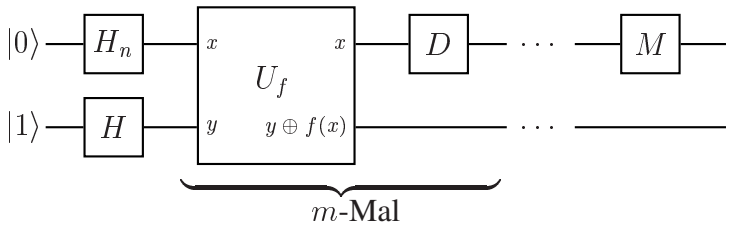
Probabilistisch: Hier liegt die erwartete Anzahl von Tests genau in der Mitte, also bei $\frac{N}{2}$.

Quantenschaltkreis: Die Behauptung ist nun, dass dieses Problem mit einem Quantenschaltkreis mit der Komplexität $O(\sqrt{N})$ gelöst werden kann, d.h. eine Verbesserung durch die Halbierung des Exponenten erreicht wird. Dieselbe Verbesserung erzielen Quantenschaltkreise auch bei NP-vollständigen Problemen, z.B. dem SAT-Problem, das an sich recht ähnlich zu dem oben geschilderten ist:
$F(x_1, ..., x_n)$ erfüllbar? $\Rightarrow f : \{0,1\}^n \rightarrow \{0,1\}$, $f(x) = 1 \Leftrightarrow x$ erfüllende

Belegung für F.
Dieses Problem kann mit einem Quantenschaltkreis in $O(2^{\frac{n}{2}})$ Schritten gelöst werden, während deterministische Algorithmen $O(2^n)$ Schritte benötigen.

4.2 Lösung mittels Quantenschaltkreis

Zur Lösung des Problems dient nun folgender Quantenschaltkreis:



Die Transformation D bedeutet dabei

$$D = \begin{pmatrix} \frac{2}{N}-1 & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N}-1 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \dots & \dots & \frac{2}{N}-1 \end{pmatrix} = H_n \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & -1 & \dots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & -1 \end{pmatrix} H_n$$

$$\text{oder auch} \quad D = \begin{pmatrix} \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \dots & \dots & \frac{2}{N} \end{pmatrix} - I$$

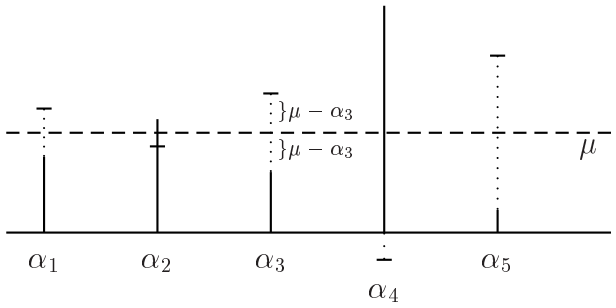
Die Anwendung von D auf einen Vektor

$$D \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{pmatrix}$$

liefert jeweils $\beta_i = \frac{2}{N} \sum_j \alpha_j - \alpha_i$. Verglichen mit dem Durchschnitt der Amplituden, der sich als

$$\mu = \frac{1}{N} \sum_{j=1}^{N} \alpha_j$$

schreiben lässt, ergibt dies $\beta_i = 2\mu - \alpha_i = \mu + (\mu - \alpha_i)$. Es entsteht also eine Umkehrung von α_i bezüglich des Durchschnittswertes μ, was an folgender Grafik verdeutlicht wird:



Beim ersten Eintritt in das U_f-Gatter findet der Übergang

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

statt, d.h. die Amplituden für alle $|z\rangle$ sind vom Betrag $\frac{1}{\sqrt{N}}$. Das negative Vorzeichen tritt dabei nur für $z = a$ auf, für alle anderen möglichen Werte sind die Amplituden positiv. Durch die oben geschilderte D-Transformation werden die Amplituden bei jedem Durchgang am aktuellen Durchschnittswert gespiegelt. Wenn wir nun mit $\alpha_x^{(t)}$ die Amplitude von $|x\rangle$ nach t Iterationen bezeichnen, so gilt

$$\alpha_a^{(0)} = -\frac{1}{\sqrt{N}} \text{ und } \alpha_x^{(0)} = \frac{1}{\sqrt{N}} \text{ für } x \neq a \text{ (vor dem ersten Durchgang)}$$

$$\mu_1 = \frac{\frac{N-1}{\sqrt{N}} - \frac{1}{\sqrt{N}}}{N} \text{ (Durchschnitt nach dem ersten Durchgang)}$$

$$\alpha_a^{(1)} = 2\mu_1 - (-\frac{1}{\sqrt{N}}) = 2\,\frac{\frac{N-1}{\sqrt{N}} - \frac{1}{\sqrt{N}}}{N} + \frac{1}{\sqrt{N}} = \frac{3}{\sqrt{N}} - \frac{4}{N\sqrt{N}} \approx \frac{3}{\sqrt{N}}$$

$$\text{und } \alpha_x^{(1)} = 2\mu_1 - \frac{1}{\sqrt{N}} = 2\,\frac{\frac{N-1}{\sqrt{N}} - \frac{1}{\sqrt{N}}}{N} - \frac{1}{\sqrt{N}} = \frac{1}{\sqrt{N}} - \frac{4}{N\sqrt{N}}.$$

Die approximative Darstellung für $\alpha_a^{(1)}$ soll nur verdeutlichen, dass die Amplitude betragsmäßig deutlich zunimmt, bei $\alpha_a^{(2)}$ erhält man bereits einen approximativen Wert von $\frac{5}{\sqrt{N}}$. Bei Fortführung der Rechnung mit exakten Werten erhält man

schließlich zwei verschränkte Rekursionsgleichungen für die α_a und α_x:

$$\begin{aligned}\alpha_a^{(t)} &= 2\mu_{t-1} + \alpha_a^{(t-1)} = (1 - \frac{2}{N})\,\alpha_a^{(t-1)} + (2 - \frac{1}{N})\,\alpha_x^{(t-1)} \\ \alpha_x^{(t)} &= 2\mu_{t-1} + \alpha_x^{(t-1)} = -\frac{1}{N}\,\alpha_a^{(t-1)} + (1 - \frac{2}{N})\,\alpha_x^{(t-1)}\end{aligned}$$

Die Lösung dieser Rekursionsgleichungen ergibt

$$\alpha_a(t) = \sin((2t+1)\theta), \quad \alpha_x(t) = \tfrac{1}{\sqrt{N-1}}\cos((2t+1)\theta),$$
$$\text{wobei} \quad \sin^2\theta = \tfrac{1}{N}.$$

Unser Ziel ist nun, $\alpha_x(t)$ für $x \neq a$ zu minimieren bzw. möglichst auf den Wert 0 zu bringen, um mit der Wahrscheinlichkeit 1 das richtige a zu messen. Wir suchen also einen Wert für t, so dass $\cos((2t+1)\theta) = 0$, was zunächst für alle t der Form $t = \frac{\pi}{4\theta} - \frac{1}{2} + m\frac{\pi}{\theta},\ m \in \mathbb{Z}$ der Fall ist.
Wählen wir beispielsweise $t_0 = \lfloor \frac{\pi}{4\theta} \rfloor$. Wenn N groß genug ist, können wir annehmen, dass $\theta \approx \sin(\theta)$. Dann gilt wegen $\sin\theta = \sqrt{\frac{1}{2^n}}$, dass

$$t_0 \approx \lfloor \tfrac{\pi}{4\sin\theta} \rfloor = \lfloor \tfrac{\pi}{4\sqrt{\frac{1}{2^n}}} \rfloor = O(\sqrt{2^n}),$$

womit die behauptete Komplexität gezeigt ist. Man kann den Algorithmus verallgemeinern für Funktionen f mit mehreren Lösungen. Sei $M = |\{x \mid f(x) = 1\}|$. Der Algorithmus von Grover kann dann ein x mit $f(x) = 1$ mit $O(\sqrt{\frac{N}{M}})$ Fragen an f finden.

4.3 Eine untere Schranke für das Problem

Zuerst beweisen wir ein technisches Lemma, das besagt, dass wenn die Wahrscheinlichkeitsverteilungen nach der Messung von zwei Zuständen $|\psi\rangle$ und $|\phi\rangle$ unterschiedlich sind, dann ist die Norm vom Differenzzustand $|\psi\rangle - |\phi\rangle$ groß.

Lemma 4.3.1 *Seien $|\psi\rangle = \sum_i \alpha_i|i\rangle$ und $|\phi\rangle = \sum_i \beta_i|i\rangle$ zwei Quantum-Zustände mit der Eigenschaft $\sum_i ||\alpha_i|^2 - |\beta_i|^2| > \epsilon$. Dann gilt $\||\psi\rangle - |\phi\rangle\| > \epsilon/2$.*

Beweis: Für den Beweis benötigen wir die Cauchy-Schwarzsche Ungleichung, die wir hier nur kurz erwähnen wollen. Sie besagt in ihrer Formulierung für Vektoren, dass für zwei Vektoren a und b gilt

$$(\sum a_i\, b_i)^2 \leq \sum a_i^2 \sum b_i^2.$$

Wir benötigen auch die Dreiecksungleichung vom Betrag: $|\alpha_i| - |\beta_i| \leq |\alpha_i - \beta_i|$.

$$\begin{aligned}
\sum_{i=1}^{N} ||\alpha_i|^2 - |\beta_i|^2| &= \sum_{i=1}^{N} (||\alpha_i| - |\beta_i||)(||\alpha_i| + |\beta_i||) \\
&\overset{\text{CSU}}{\leq} \sqrt{\sum_{i=1}^{N} (|\alpha_i| - |\beta_i|)^2} \sqrt{\sum_{i=1}^{N} (|\alpha_i| + |\beta_i|)^2} \\
&\overset{\text{DU}}{\leq} \sqrt{\sum_{i=1}^{N} (|\alpha_i - \beta_i|)^2} \sqrt{\sum_{i=1}^{N} (|\alpha_i| + |\beta_i|)^2} \\
&= \||\psi\rangle - |\phi\rangle\| \sqrt{\sum_{i=1}^{N} (|\alpha_i|^2 + |\beta_i|^2 + 2|\alpha_i||\beta_i|)} \\
&\overset{\text{CSU}}{\leq} \||\psi\rangle - |\phi\rangle\| \sqrt{2 + 2\sqrt{\sum_{i=1}^{N} |\alpha_i|^2 \sum_{i=1}^{N} |\beta_i|^2}} \\
&= 2\||\psi\rangle - |\phi\rangle\|.
\end{aligned}$$

□

Wir betrachten nun einen Quantenalgorithmus A, der bildlich gesehen Fragen an U_f stellen kann (weshalb im folgenden dafür auch die Abkürzung O wie Orakel stehen soll). Die Behauptung ist nun, dass $A\, \Omega(\sqrt{N})$ Fragen benötigt, um a zu finden. Wir stellen uns den Algorithmus so vor, dass er immer abwechselnd eine Transformation vornimmt und eine Frage stellt, wobei die Gesamtzahl an Fragen mit T festgelegt ist:

$$\underbrace{U_{T-1}\, O \ldots O\, U_2\, O\, U_1\, O\, U_0}_{A}\, |0^n\rangle$$

Sei O_f das Orakel für die Funktion f definiert als $f(z) = 0, \forall z$. Sei nun $|\psi_t\rangle$ der Zustand vor der Frage t, d.h. $|\psi_t\rangle = U_{t-1}\, O_f \ldots O_f\, U_2\, O_f\, U_1\, O_f\, U_0\, |0^n\rangle$. Dieser Zustand lässt sich auch in Amplitudendarstellung schreiben als $|\psi_t\rangle = (\alpha_0\, |0\rangle + \alpha_1\, |1\rangle + \ldots + \alpha_i\, |a\rangle + \ldots)$.
Sei nun $q_x(\psi_t) = |\alpha_x|^2$. Dann gilt natürlich

$$\sum_{t=1}^{T} \underbrace{\sum_{x \in \{0,1\}^n} q_x(\psi_t)}_{1} = T$$

Durch Vertauschen der Summationsreihenfolge lässt sich aus dieser Aussage eine weitere Aussage gewinnen:

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

Wir definieren jetzt eine neue Funktion g, als

$$g(x) = \begin{cases} 1 & \text{für } x = x_0 \\ 0 & \text{sonst} \end{cases}$$

Lassen wir nun A mit f und dem zugehörigen Orakel O_f laufen, so erhalten wir

$$|\psi_T\rangle = U_{T-1}\, O_f \ldots U_1\, O_f\, U_0\, |0\rangle,$$

Wenden wir A hingegen auf g mit dem Orakel O_g an, so erhalten wir

$$|\psi'_T\rangle = U_{T-1}\, O_g \ldots U_1\, O_g\, U_0\, |0\rangle$$

Nach einer Messung sollte man in der Lage sein, $|\psi_T\rangle$ von $|\psi'_T\rangle$ zu unterscheiden. Daraus folgt, dass die Wahrscheinlichkeitsverteilung nach der Messung von $|\psi_T\rangle$ weit von der Verteilung nach der Messung von $|\psi'_T\rangle$ entfernt sein soll (Aufgabe 9). Das impliziert (Lemma 4.3.1), dass die Norm der Differenz $\| \, |\psi_T\rangle - |\psi'_T\rangle \, \|$ groß sein soll. Wir legen fest, dass die Differenz $\| |\psi_T\rangle - |\psi'_T\rangle \| \geq \frac{1}{2}$ sein soll, um mit höherer Wahrscheinlichkeit $|\psi_T\rangle$ von $|\psi'_T\rangle$ unterscheiden zu können.
Wir definieren nun noch

$$|\psi_T\rangle_i = U_{T-1}\, O_g \ldots O_g\, U_i\, O_f\, U_{i-1} \ldots O_f\, U_0 |0\rangle,$$

wobei klar ist, dass $|\psi_T\rangle_0 = |\psi'_T\rangle$ und $|\psi_T\rangle_T = |\psi_T\rangle$. Damit gilt dann

$$\begin{aligned}
\| \, |\psi_T\rangle - |\psi'_T\rangle \| &= \| \, |\psi_T\rangle_T - |\psi_T\rangle_0 \| \\
&= \| \, |\psi_T\rangle_T - |\psi_T\rangle_1 + |\psi_T\rangle_1 - |\psi_T\rangle_0 \| \\
&= \| \sum_{i=1}^{T} (|\psi_T\rangle_i - |\psi_T\rangle_{i-1}) \| \\
&\overset{\Delta\text{-Ungl.}}{\leq} \sum_{i=1}^{T} \| \, |\psi_T\rangle_i - |\psi_T\rangle_{i-1} \| \\
&= \sum_{i=1}^{T} \|E_i\|
\end{aligned}$$

Zur Vereinfachung schreiben wir nun

$$\begin{aligned} |\psi_T\rangle_i &= \overbrace{U_{T-1}\, O_g \ldots O_g\, U_i}^{U}\, O_f\, \overbrace{U_{i-1} \ldots O_f\, U_0\, |0\rangle}^{|\psi_i\rangle} \\ |\psi_T\rangle_{i-1} &= \underbrace{U_{T-1}\, O_g \ldots O_g\, U_i}_{U}\, O_g\, U_{i-1} \ldots O_f\, U_0\, |0\rangle \end{aligned}$$

Damit gilt

$$\begin{aligned} |\psi_T\rangle_i &= U\, O_f\, |\psi_i\rangle = U\, |\psi_i\rangle \\ |\psi_T\rangle_{i-1} &= U\, O_g\, |\psi_i\rangle = U\, (|\psi_i\rangle - 2\, \alpha_{x_0}^{(i)}\, |x_0\rangle) \end{aligned}$$

(O_g transformiert $|i\rangle$ in $(-1)^{g(i)}|i\rangle$) und somit

$$\begin{aligned} \|E_i\| &= \|U|\psi_i\rangle - U\, (|\psi_i\rangle - 2\, \alpha_{x_0}^{(i)}\, |x_0\rangle)\| = \|U\, 2\, \alpha_x^{(i)}\, |x_0\rangle\| \\ &\overset{U\text{unitär}}{=} \|2\, \alpha_{x_0}^{(i)}\, |x_0\rangle\| = \sqrt{4\, |\alpha_{x_0}^{(i)}|^2} = 2\, |\alpha_{x_0}^{(i)}| \end{aligned}$$

Für den nächsten Schritt benötigen wir wieder die Cauchy-Schwarzsche Ungleichung. Damit können wir nun für die Summe der E_i schreiben

$$\begin{aligned} \sum_{i=1}^{T} \|E_i\| &= \sum_{i=1}^{T} \|E_i\|\, 1 \overset{\text{CSU}}{\leq} \sqrt{\sum_{i=1}^{T} \|E_i\|^2 \sum_{i=1}^{T} 1^2} \\ &= \sqrt{\sum_{i=1}^{T} \|E_i\|^2\, T} = \sqrt{\sum_{i=1}^{T} 4\, |\alpha_{x_0}^{(i)}|^2\, T} \\ &= 2 \sqrt{\sum_{i} q_{x_0}(\psi_i)\, T} \overset{\text{s.o.}}{\leq} 2 \sqrt{\frac{T}{N}\, T} = 2\, \frac{T}{\sqrt{N}} \end{aligned}$$

Insgesamt können wir jetzt zwei Bedingungen für $\| |\psi_T\rangle - |\psi_T'\rangle \|$ gegenüberstellen:

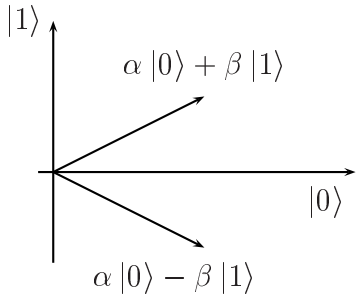
$$\| \, |\psi_T\rangle - |\psi_T'\rangle \| \leq 2\, \tfrac{T}{\sqrt{N}} \quad \text{und} \quad \| |\psi_T\rangle - |\psi_T'\rangle \| \geq \tfrac{1}{2},$$

woraus sich für T die Bedingung $T \geq \frac{\sqrt{N}}{4} = \Omega(\sqrt{N})$ ergibt.

4.4 Geometrische Interpretation

Betrachten wir zunächst eine einfache Analogie in der Zahlenebene mit Basis $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Wenn wir uns darin den Vektor $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha\, |0\rangle + \beta\, |1\rangle$

vorstellen und auf ihn die Transformation $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ anwenden, so erhalten wir den Vektor $\begin{pmatrix} \alpha \\ -\beta \end{pmatrix}$, was im folgenden veranschaulicht wird.



Bildlich veranschaulicht lässt sich leicht einsehen, dass man eine solche Transformation auch als Reflexion um $|0\rangle$ bezeichnet, da der Vektor an der Achse gespiegelt wird.
Betrachten wir nun den Fall in höheren Dimensionen, so kann man hier in gewissen Fällen ebenfalls von einer Reflexion sprechen. Die Transformation D des Grover-Suchalgorithmus

$$D = H_n \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & -1 & \dots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & -1 \end{pmatrix} H_n$$

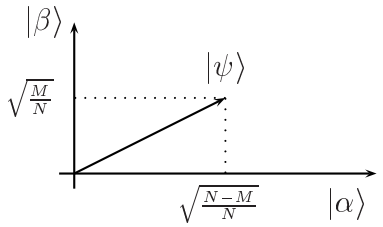
bewirkt aufgrund der -1 Faktoren eine Reflexion um $|0^n\rangle$. (H_n tansformiert $|0^n\rangle$ in $|\psi\rangle$ und zurück).
Betrachten wir nun eine allgemeine Funktion $f : \{0,1\}^n \to \{0,1\}$ und bezeichnen wieder $N = 2^n$. Ferner sei $M = |\{x \in \{0,1\}^n \mid f(x) = 1\}|$, d.h. die Mächtigkeit der Menge aller Eingabevektoren, für die die Funktion den Wert 1 liefert. Wir legen nun zwei Elemente fest:

$$|\beta\rangle = \tfrac{1}{\sqrt{M}} \sum_{\{x|f(x)=1\}} |x\rangle \quad \text{und} \quad |\alpha\rangle = \tfrac{1}{\sqrt{N-M}} \sum_{\{x|f(x)=0\}} |x\rangle$$

Durch Nachrechnen kann man leicht überprüfen, dass $\langle\alpha|\beta\rangle = 0$, d.h. die Elemente sind orthogonal zueinander. Wenn wir noch einmal die erste Transformation aus dem Grover-Algorithmus heranziehen (die Anwendung von H_n auf $|0^n\rangle$, die als Ergebnis $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_i |i\rangle$ liefert), so können wir das Ergebnis dieser Transformation auch darstellen als

$$|\psi\rangle = \sqrt{\tfrac{M}{N}}\,|\beta\rangle + \sqrt{\tfrac{N-M}{N}}\,|\alpha\rangle$$

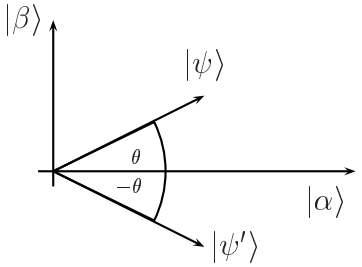
und es folgendermaßen geometrisch veranschaulichen:



Wenden wir nun die Orakelfunktion O_f auf $|\psi\rangle$ an, so dreht sich bei den Eingabewerten, für die $f(x) = 1$ ist, das Vorzeichen um und wir erhalten

$$O_f\,|\psi\rangle = -\sqrt{\tfrac{M}{N}}\,|\beta\rangle + \sqrt{\tfrac{N-M}{N}}\,|\alpha\rangle = |\psi'\rangle,$$

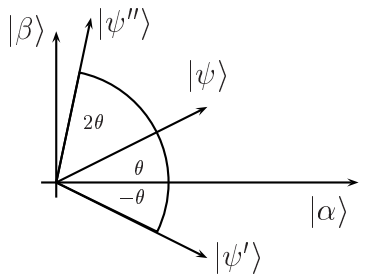
was als eine Reflexion von $|\psi\rangle$ an α betrachtet werden kann:



Bei der folgenden Transformation D haben wir es wieder mit einer Reflexion zu tun, diesmal jedoch an $|\psi\rangle$:

$$H_n \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & -1 & \dots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & -1 \end{pmatrix} H_n\,|\psi'\rangle = |\psi''\rangle$$

Zwei Reflexionen ergeben zusammen eine Rotation, und man erhält als geometrische Darstellung



In Bezug auf den realen Algorithmus ist der Winkel θ natürlich wesentlich kleiner als auf den obigen Schaubildern, auf denen nur zur Verdeutlichung der Winkel stark übertrieben eingezeichnet wurde. Insbesondere wird klar, dass wegen $\cos\theta = \sqrt{\frac{N-M}{N}}$ und $\sin\theta = \sqrt{\frac{M}{N}}$ für $M = 1$ ein sehr geringer Winkel vorherrscht. Dieser Winkel kann nun aber durch wiederholte Reflexionen bzw. Rotationen wie im Schaubild angedeutet verstärkt werden. Man erhält für die wiederholte Anwendung der Transformation $G = D\,O_f$ auf $|\psi\rangle = \cos\theta\,|\alpha\rangle + \sin\theta|\beta\rangle$

$$G\,|\psi\rangle = \cos 3\theta\,|\alpha\rangle + \sin 3\theta\,|\beta\rangle$$
$$G^t\,|\psi\rangle = \cos(2t+1)\theta\,|\alpha\rangle + \sin(2t+1)\theta\,|\beta\rangle$$

Für $M = 1$ wollen wir, dass der Ausdruck $(2t+1)\theta \approx (2t+1)\sqrt{\frac{1}{N}}$ den Wert $\frac{\pi}{2}$ annimmt und erhalten somit für $t \approx \frac{\pi}{4}\sqrt{N}$. Diese Abschätzung hatten wir auch mit der numerischen Herleitung erhalten.

4.5 Anwendungen des Algorithmus

Der Suchalgorithmus von Grover lässt sich auch in verallgemeinerter Form für Funktionen formulieren, die für mehrere Eingaben den Wert 1 liefern. Wenn man bei dem Bild mit der Nadel im Heuhaufen bleibt, so entspricht dies mehreren Nadeln in einem Heuhaufen.

Behauptung 4.5.1 (QSEARCH) *Sei $f : \{0,1\}^n \to \{0,1\}$ mit einem Orakelschaltkreis U_f und $M = \| \{x \mid f(x) = 1\} \|$.*
Dann gibt es einen Quantenalgorithmus QSEARCH mit der Eigenschaft:

- *Falls $M \neq 0$ findet QSEARCH(f) ein x mit $f(x) = 1$ nach einer erwarteten Anzahl von $O(\sqrt{N/M})$ Fragen an U_f. Alle solche Lösungen haben die gleiche Wahrscheinlichkeit gefunden zu werden.*

- *Falls $M = 0$ hält QSEARCH(f) nicht.*

Die Begründung für die Behauptung ist der Beweis zum Algorithmus von Grover. Im folgenden werden verschiedene Beispiele für konkrete Anwendungen eines solchen QSEARCH-Algorithmus betrachtet.

4.5.1 Das Minimum-Problem

Gegeben sei eine injektive Funktion $f : A \to B$, wobei die Menge A der Einfachheit halber als $A = \{0, 1, ..., N-1\}$ gewählt sei. Das Minimum-Problem besteht darin, ein $i \in A$ zu finden, so dass $f(i)$ ein Minimum darstellt, d.h. es muss gelten $f(j) > f(i)\, \forall j \neq i$.
Die Frage ist nun wieder, wieviele Vergleiche von Funktionswerten zur Ermittlung des Minimums angestellt werden müssen.
Es ist klar, dass im deterministischen Fall $N-1$ Vergleiche angestellt werden müssen. Probabilistische Algorithmen brauchen auch linear viele Vergleiche. Allerdings lässt sich unter Zulassung einer gewissen Fehlerwahrscheinlichkeit mit einem Quantenalgorithmus eine Zeit von $O(\sqrt{N})$ erreichen.
Wir werden zuerst einen Algorithmus für das Minimum beschreiben mit einer erwarteten Laufzeit von $O(\sqrt{N})$. Der Algorithmus lässt sich einfach transformieren in einen probabilistischen Algorithmus mit Laufzeit $O(\sqrt{N})$. Die erwartete Zeit eines Algorithmus erhält man, indem man die Laufzeit als Zufallsvariable betrachtet und ihren Erwartungswert ausrechnet. Ein einfaches Beispiel ist der Algorithmus

```
repeat
  Wähle i ∈R {0, 1}
until i=0 ,
```

wobei das Symbol $\in_R$ die zufällige, gleichverteilte Auswahl aus der darauffolgenden Menge bedeutet.
Wir betrachten die erwartete Anzahl an Schleifendurchläufen: Ein Durchgang der Schleife findet auf jeden Fall statt, der zweite nur noch mit Wahrscheinlichkeit $\frac{1}{2}$, der dritte mit Wahrscheinlichkeit $\frac{1}{4}$ usw. Für den Erwartungswert erhält man dann

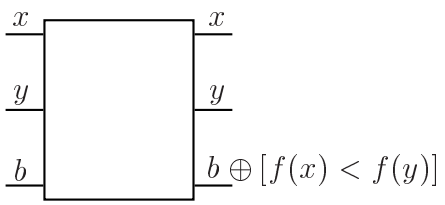
$$E = \sum_{i=1}^{\infty} i \cdot \text{Prob[Der Algorithmus macht genau } i \text{ Schleifendurchläufe]} = \sum_{i=1}^{\infty} i \cdot 2^{-i} = 2$$

Damit endet dieser Algorithmus mit einer erwarteten Anzahl von 2 Schleifendurchläufen.

Modellierung des Problems: Durch die Definition einer Funktion der Gestalt

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

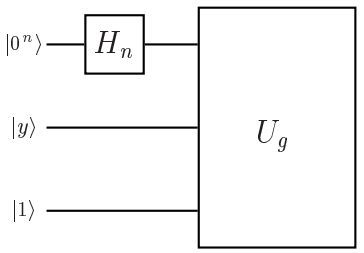
was z.B. mit einem Gatter der Form



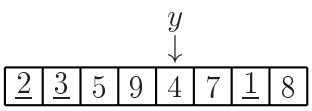
vorstellbar ist (wie genau der Vergleich stattfindet, sei an dieser Stelle irrelevant), kann man mit QSEARCH(g) passende Werte für x und y finden, so dass gilt $g(x,y) = 1 \Leftrightarrow f(x) < f(y)$. Damit lässt sich ein Quantenalgorithmus für das Problem angeben:

Wähle $y \in_R A$
`repeat`
    $|\phi\rangle := (H_n|0^n\rangle)|y\rangle|1\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle|y\rangle|1\rangle$
    QSEARCH($g(x,y)$)
    y' := Messung der ersten n Register
    `if` $f(y') < f(y)$ `then` $y := y'$
`until` Anzahl von Fragen an $f > l$
Ausgabe $(y, f(y))$

l ist ein Parameter, der später fixiert wird. Der schematische Schaltkreis zu diesem Algorithmus wäre also



Man beachte auch, dass im Algorithmus die Anzahl von Fragen an f als Abbruchbedingung der Schleife ausgewertet wird, damit der Fall der Unendlichschleife in QSEARCH berücksichtigt wird. Die erste Hadamard-Transformation produziert einen Zustand $|\psi\rangle$, in dem man Fragen der Art $[f(x) < f(y)]$ stellen kann. Beispielhaft sei das Vorgehen des Algorithmus hier an einer einfachen Skizze verdeutlicht, wobei die Position im Array das Argument, die darinstehende Zahl den Funktionswert repräsentiert:



Der Algorithmus findet bei gegebenem y mit jeweils gleicher Wahrscheinlichkeit einen der möglichen Werte für x, so dass $f(x) < f(y)$ (in der Skizze ist das für alle unterstrichenen Elemente der Fall).
Wir wollen nun die Laufzeit dieses Quantenalgorithmus bestimmen, wozu jedoch zunächst einige Vorüberlegungen notwendig sind. Wir nehmen zuerst an, dass l unendlich ist. Es sollte klar sein, dass der Algorithmus immer das Minimum findet, wenn l groß genug ist.

Lemma 4.5.2 *Sei $p(t, r)$ die Wahrscheinlichkeit, dass der Minimum-Algorithmus irgendwann während seiner Ausführung den Index des r-kleinsten Elementes (in sortierter Reihenfolge) wählt, wenn er in einer Menge von t Elementen sucht. Dann gilt*

$$p(t,r) = \begin{cases} \frac{1}{r} & \textit{falls } r \leq t \\ 0 & \textit{sonst} \end{cases}$$

Beweis: Der Beweis zu diesem Lemma kann mittels vollständiger Induktion über t erfolgen.
Basisfall $t = r$: Das r-kleinste Element kann nur bei der ersten Auswahl gewählt werden, da für jede andere Wahl des ersten Elementes ein kleineres Element gewählt wird und somit im folgenden jeweils nur noch kleinere Werte in Frage kommen. Bei der ersten Auswahl wird es unter Gleichverteilung mit Wahrscheinlichkeit $\frac{1}{r}$ gewählt.
Induktionsschritt: Man betrachte $p(t + 1, r)$. Diese Wahrscheinlichkeit setzt sich zusammen aus der Wahrscheinlichkeit, dass das r-te Element direkt gewählt wurde ($p = \frac{1}{t+1}$) und der Summe der Wahrscheinlichkeiten, dass zunächst ein größeres Element gewählt wurde, wodurch die Möglichkeit einer späteren Wahl offen bleibt.

Damit ergibt sich

$$\begin{aligned} p(t+1,r) &= \frac{1}{t+1} + \sum_{k=r+1}^{t+1} \frac{1}{t+1}\, p(k-1,r) \\ &\overset{IH}{=} \frac{1}{t+1} + \sum_{k=r+1}^{t+1} \frac{1}{t+1}\,\frac{1}{r} \\ &= \frac{1}{t+1}\,(1+\frac{1}{r}(t+1-r)) \\ &= \frac{1}{t+1}\,(\frac{t+1}{r}) = \frac{1}{r} \end{aligned}$$

□

Lemma 4.5.3 *Die erwartete Anzahl von Fragen an f, bis der Wert y dem Minimum entspricht, ist $O(\sqrt{N})$.*

Beweis: Wir wir gesehen haben, ist die erwartete Anzahl von Fragen an den QSEARCH-Algorithmus $O(\sqrt{\frac{N}{M}})$, für $r-1$ Elemente also $\leq c\,\sqrt{\frac{N}{r-1}}$ für eine Konstante c. Für den Erwartungswert der Fragenanzahl müssen wir jeweils die Wahrscheinlichkeit, dass das r-te Element irgendwann als y gewählt wird, mit der Anzahl von Fragen multiplizieren, die QSEARCH benötigt, um ein Element x mit $f(x) < f(y)$ aus den verbleibenden $r-1$ Elementen zu finden; diese Werte müssen dann für $r = 2$ bis $r = N$ aufsummiert werden. Mit Hilfe des vorigen Lemmas erhalten wir damit folgende Abschätzung:

$$\begin{aligned} E[\text{Fragenanzahl}] &= \sum_{r=2}^{N} p(N,r) \cdot [\text{Anzahl der Fragen an QSEARCH}] \\ &\leq \sum_{r=2}^{N} p(N,r) \cdot c\,\sqrt{\frac{N}{r-1}} = c\,\sqrt{N} \sum_{r=1}^{N-1} \frac{1}{r+1}\,\frac{1}{\sqrt{r}} \\ &\leq c\,\sqrt{N}\,(\frac{1}{2} + \sum_{r=1}^{N-1} r^{-\frac{3}{2}}) \end{aligned}$$

Die letzte Summe kann durch ein Integral nach oben abschätzt werden und man erhält weiter:

$$E[\text{Fragenanzahl}] \leq c\,\sqrt{N}\,(\tfrac{1}{2} + \int_{1}^{N-1} r^{-\frac{3}{2}} dr) = c\,\sqrt{N}\,(\tfrac{1}{2} + 2 - \tfrac{2}{\sqrt{N-1}}) \leq c\,\sqrt{N}\,\tfrac{5}{2} = O(\sqrt{N})$$

Wählt man nun $l = 5\,c\,\sqrt{N}$, so findet der Algorithmus das Minimum mit Wahrscheinlichkeit $> \frac{1}{2}$, da sonst gelten würde

$$\begin{aligned} E[\text{Fragenanzahl}] &= \sum_{i=1}^{\infty} i\,Prob[i \text{ Fragen werden gestellt bis das Minimum gefunden wird}] \\ &= \sum_{i=1}^{l} i\,Prob[...] + \sum_{i=l+1}^{\infty} i\,Prob[...] \geq (l+1)\,\frac{1}{2} > \frac{l}{2} \geq E[\text{Fragenanzahl}] \end{aligned}$$

Durch das Größer-Zeichen entsteht ein Widerspruch, also muss die Wahrscheinlichkeit, dass weniger als l Fragen gestellt werden, bis das Minimum gefunden wird, größer als $\frac{1}{2}$ sein. □

Untere Schranke für das Minimum-Problem: Wir führen das Minimum-Problem durch eine Art Reduktion auf den Grover-Suchalgorithmus zurück und zeigen so, dass für dieses Problem dieselbe untere Schranke gelten muss. Dazu konstruieren wir eine Funktion $g : \{0, ..., N-1\} \rightarrow \{0, ..., 2N-1\}$ mit $g(i) = i + N(1 - f(i))$, d.h.

$$g(i) = \begin{cases} i + N & \text{falls } f(i) = 0 \\ i & \text{falls } f(i) = 1 \end{cases}$$

Es gelten nun die Zusammenhänge

$$\begin{aligned} \forall\, x\; f(x) = 0 &\Rightarrow Min(g) = N \\ \exists\, y :\; f(y) = 1 &\Rightarrow Min(g) < N \end{aligned}$$

Ein Algorithmus für das Minimum-Problem auf g muss somit ein i finden mit $f(i) = 1$, und leistet damit dasselbe, was auch der Grover-Algorithmus bewirken würde; für diesen galt jedoch eine untere Schranke von $\Omega(\sqrt{N})$, die deshalb auch für das Minimum-Problem gelten muss.

4.5.2 Element distinctness

Gegeben sei eine Funktion $f : \{0, ..., N-1\} \rightarrow \{0, ..., M-1\}$. Ziel ist es nun, Elemente x und y mit $x \neq y$ zu finden, so dass $f(x) = f(y)$ (falls es solche Elemente gibt, d.h. die Funktion nicht injektiv ist).

Deterministische Lösung: Die naheliegende Methode wäre, zunächst alle Funktionswerte zu sortieren und anschließend in der sortierten Liste nach doppelten Einträgen zu suchen, was auf eine Laufzeit von $O(N \log N) + O(N) = O(N \log N)$ hinauslaufen würde.

Quantenalgorithmus: Ein Quantenalgorithmus für dieses Problem könnte folgendermaßen vorgehen:

1. Wähle eine zufällige Teilmenge $A \subseteq_R [N]$ mit l Elementen
2. Wähle eine zufällige Teilmenge $B \subseteq_R [N]$ mit l^2 Elementen
3. Sortiere die Elemente in A nach Funktions-Wert von f
4. "Quantum search" für ein $b \in B$, so dass es ein $a \in A$, $a \neq b$ mit $f(a) = f(b)$ gibt
5. Amplifiziere die Amplituden in 1–4

Betrachten wir nun die Komplexität dieses Algorithmus: In den ersten beiden Schritten wird einfach mit $O(l)$ und $O(l^2)$ eine Auswahl getroffen. Im dritten Schritt kann mit einem geeigneten Sortieralgorithmus mit Komplexität $O(l \log l)$ sortiert werden. Im vierten Schritt schließlich ist für ein festes $b \in B$ wegen der Sortierung von A nach Funktionswerten nur eine Binärsuche nach $f(b)$ notwendig (mit einem zusätzlichen Vergleich, ob $a \neq b$), was in $O(\log l)$ liegt. Allerdings muss ein geeignetes b erst gefunden werden, was durch den QSEARCH-Algorithmus geschieht, der dann die Komplexität $O(\sqrt{|B|})$ hat. Da der QSEARCH-Algorithmus in jedem Schritt zur Verifizierung die Binärsuche durchführen muss, erhält man insgesamt für den vierten Schritt eine Komplexität von $O(\sqrt{|B|}) \log l = O(l \log l)$. Für gegebene Mengen A und B terminiert der Algorithmus also in $O(l \log l)$.
Es ist allerdings nicht sicher, ob die beiden Mengen A und B zufällig so gewählt wurden, dass zwei Elemente x und y, $x \neq y$ mit $f(x) = f(y)$ gefunden werden. Wenn wir annehmen, dass es nur einen solchen Fall gibt, so ist die Anzahl möglicher Auswahlen A, die x enthalten $\binom{N-1}{l-1}$; die Anzahl möglicher Auswahlen B, die y enthalten, ist dementsprechend $\binom{N-1}{l^2-1}$.
Die Anwendung des Grover-Algorithmus auf alle möglichen Paare A und B (das geschieht in der Amplifizierung im Schritt 5) liefert dann eine Komplexität von

$$O(\sqrt{\frac{\binom{N}{l}\binom{N}{l^2}}{\binom{N-1}{l-1}\binom{N-1}{l^2-1}}}) = O(\sqrt{\frac{N}{l}\frac{N}{l^2}}) = O(\frac{N}{\sqrt{l^3}})$$

Somit sind ingesamt $O(\frac{N}{\sqrt{l}} \log l)$ Fragen notwendig. Wählen wir $l = \sqrt{N}$, so ist dies $O(N^{\frac{3}{4}} \log \sqrt{N})$, was einiges über der unteren Schranke für dieses Problem liegt, die $\Omega(N^{\frac{1}{2}})$ beträgt.

Kapitel 5

Algorithmus von Shor für die Faktorisierung

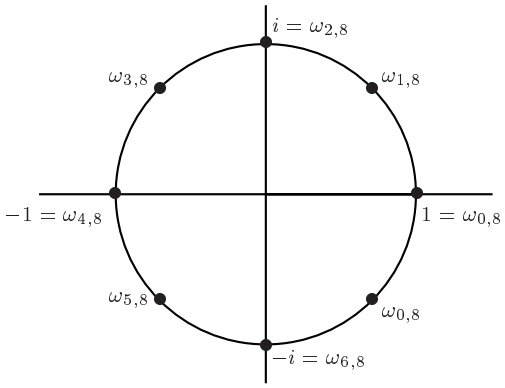
Ziel dieser Vorlesungseinheit ist es, möglichst verständlich und umfassend den Quantenalgorithmus von Shor zur Faktorisierung von Zahlen einzuführen. Dazu sind allerdings eine Reihe von Vorüberlegungen und Grundlagen erforderlich, die im folgenden aufgeführt sind.

5.1 Diskrete Fourier-Transformation

Die diskrete Fouriertransformation (DFT) ist folgendermaßen definiert: Gegeben sei ein Vektor $x = (x_0, ..., x_{N-1})$. Dann gilt für seine Transformierte $DFT(x) = (y_0, ..., y_{N-1})$ mit den Koeffizienten

$$y_k = \sum_{j=0}^{N-1} e^{\frac{2\pi ijk}{N}} x_j = \sum_{j=0}^{N-1} x_j \, (\omega_{j,N})^k.$$

Dabei sind die $(\omega_{j,N}) = e^{2\pi ij/N}$ die N-ten Einheitswurzeln, die sich beispielsweise für $N = 8$ wie in der folgenden Darstellung im Einheitskreis der komplexen Zahlenebene darstellen lassen:



Rein rechnerisch kann man mit der Eulerschen Formel auch leicht nachvollziehen, dass $(\omega_{j,N})^N = e^{2\pi ij} = \cos(2\pi j) + i\ \sin(2\pi j) = 1$. Im Rahmen der Vorüberlegung wollen wir auch noch eine Reihe nützlicher Zusammenhänge betrachten:

$$\begin{aligned}
(\omega_{k,N})^2 &= e^{\frac{ik2\pi}{N/2}} = \omega_{k,N/2} \text{ für } k = 0, ..., \frac{N}{2} - 1 \\
(\omega_{\frac{N}{2}+k,N})^2 &= e^{\frac{i(\frac{N}{2}+k)2\pi}{N/2}} = e^{i2\pi}\, e^{\frac{ik2\pi}{N/2}} = \omega_{k,N/2} \text{ für } k = \frac{N}{2}, ..., N \\
\omega_{j,N} &= e^{\frac{2\pi ij}{N}} = (e^{\frac{2\pi i}{N}})^j = (\omega_{1,N})^j \\
\omega_{2j,N} &= \omega_{j,N}^2 = \omega_{j,N/2}
\end{aligned}$$

Für die Indizes k der Fouriertransformation erhält man so die Werte $1, \omega_{1,N}, \omega_{1,N}^2, ..., \omega_{1,N}^{N-1}$. Damit kann die Transformation mit der vereinfachten Schreibweise $\omega_{1,N} = z$ auch als Matrix geschrieben werden:

$$\begin{pmatrix} y_0 \\ \vdots \\ \vdots \\ y_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & \dots & 1 \\ 1 & z & z^2 & \dots & z^{N-1} \\ 1 & z^2 & z^4 & \dots & z^{2(N-1)} \\ 1 & z^3 & z^6 & \dots & z^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z^{N-1} & z^{2(N-1)} & \dots & z^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ \vdots \\ x_{N-1} \end{pmatrix}$$

5.2 Fast Fourier Transformation

Die diskrete Fourier-Transformation benötigt viele teilweise redundante Berechnungsschritte, wenn sie nach dem naiven Schema wie oben dargestellt ausgeführt wird. Eine deutliche Effizienzsteigerung wird durch den Algorithmus von Cooley/Tukey (1965) erreicht, der als schnelle Fourier-Transformation (FFT) bezeichnet wird; er erreicht eine Komplexität von $O(N\ \log N)$.

Für das Verfahren wird zunächst der Vektor $x = (x_0, ..., x_{N-1})$ (der hier eine gerade Anzahl an Elementen haben soll) in die zwei Vektoren $x^{even} = (x_0, x_2, ..., x_{N-2})$ und $x^{odd} = (x_1, x_3, ..., x_{N-1})$ aufgespalten. Anschließend wird eine erneute FFT auf x^{even} und x^{odd} angewendet, so dass man als Ergebnis $y^{even} = (y_0^{even}, y_1^{even}, ..., y_{\frac{N}{2}-1}^{even})$ und $y^{odd} = (y_0^{odd}, y_1^{odd}, ..., y_{\frac{N}{2}-1}^{odd})$ erhält. Die Koeffizienten sind im Detail

$$\begin{aligned} y_k^{even} &= \sum_{j=0}^{\frac{N}{2}-1} x_{2j}\, e^{\frac{ijk2\pi}{N/2}} = \sum_{j=0}^{\frac{N}{2}-1} x_{2j}\, (\omega_{j,N})^{2k} = \sum_{j=0}^{\frac{N}{2}-1} x_{2j}\, (\omega_{j,\frac{N}{2}})^k \\ y_k^{odd} &= \sum_{j=0}^{\frac{N}{2}-1} x_{2j+1}\, e^{\frac{ijk2\pi}{N/2}} = \sum_{j=0}^{\frac{N}{2}-1} x_{2j+1}\, (\omega_{j,\frac{N}{2}})^k \end{aligned}$$

Nun müssen die einzelnen Koeffizienten in geeigneter Weise wieder zusammengeführt werden, um den Lösungsvektor $y = (y_0, y_1, ..., y_{N-1})$ der Fourier-Transformation zu erhalten. Dazu betrachten wir zunächst den Fall $k = 0, ..., \frac{N}{2} - 1$:

$$\begin{aligned} y_k &= \sum_{j=0}^{N-1} x_j\, (\omega_{j,N})^k \\ &= \sum_{j=0}^{\frac{N}{2}-1} x_{2j}\, (\omega_{2j,N})^k + \sum_{j=0}^{\frac{N}{2}-1} x_{2j+1}\, (\omega_{2j+1,N})^k \\ &= \sum_{j=0}^{\frac{N}{2}-1} x_{2j}\, (\omega_{j,\frac{N}{2}})^k + \sum_{j=0}^{\frac{N}{2}-1} x_{2j+1}\, (\omega_{j,\frac{N}{2}})^k\, \omega_{k,N} \\ &= y_k^{even} + y_k^{odd}\, \omega_{k,N} \end{aligned}$$

Die dritte Gleichheit folgt weil $(\omega_{2j+1,N})^k = (\omega_{1,N})^{(2j+1)k} = (\omega_{1,N})^{2jk} (\omega_{1,N})^k = (\omega_{j,\frac{N}{2}})^k\, \omega_{k,N}$. Auf ähnliche Weise erhält man für $l = \frac{N}{2}, ..., N-1$ und $l = \frac{N}{2} + k$:

$$\begin{aligned} y_{\frac{N}{2}+k} &= \sum_{j=0}^{N-1} x_j\, (\omega_{j,N})^{\frac{N}{2}+k} \\ &= \sum_{j=0}^{\frac{N}{2}-1} x_{2j}\, (\omega_{2j,N})^{\frac{N}{2}+k} + \sum_{j=0}^{\frac{N}{2}-1} x_{2j+1}\, (\omega_{2j+1,N})^{\frac{N}{2}+k} \\ &= \sum_{j=0}^{\frac{N}{2}-1} x_{2j}\, (\omega_{2j,N})^k (\omega_{2j,N})^{\frac{N}{2}} + \sum_{j=0}^{\frac{N}{2}-1} x_{2j+1}\, (\omega_{j,\frac{N}{2}})^k\, \omega_{k,N}\, \underbrace{(\omega_{1,N})^{\frac{N}{2}}}_{=-1} \\ &= y_k^{even} - y_k^{odd}\, \omega_{k,N} \end{aligned}$$

Insgesamt ergibt sich also der rekursive Algorithmus

```
Procedure FFT(N, x_0 ..., x_{N-1})
if N = 1 then return (x_0)
    else
    x^even := (x_0, x_2, ..., x_{N-2})
    x^odd := (x_1, x_3, ..., x_{N-1})
    y^even := FFT(N/2, x^even)
    y^odd := FFT(N/2, x^odd)
    z := 1
    u := e^{i2π/N}
    for k := 0 to N/2 - 1 do
        y_k := y_k^even + z * y_k^odd
        y_{N/2+k} := y_k^even - z * y_k^odd
        z := z * u
    return(y_0, ..., y_{N-1})
```

Für die Komplexität erhält man damit die Rekursionsgleichung $T(N) \leq 2T(\frac{N}{2}) + O(N) = O(N \log N)$.

5.3 Quantum Fourier Transformation

Bei der Quantum Fourier Transformation (QFT) wird ein Qubit $|j\rangle$ in einen Überlagerungszustand transformiert:

$$|j\rangle \overset{QFT}{\rightarrow} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle$$

Somit wird eine Linearkombination $\sum_{j=0}^{N-1} x_j |j\rangle$ in eine neue Linearkombination transformiert, die man mit Hilfe der klassischen DFT erhält:

$$\begin{aligned}
\sum_{j=0}^{N-1} x_j |j\rangle &\rightarrow \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} x_j \, e^{\frac{2\pi i j k}{N}} |k\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k |k\rangle ,
\end{aligned}$$

wobei $(y_0, ..., y_{N-1}) = DFT(x_0, ..., x_{N-1})$.
Durch Anwendung der oben erwähnten Matrix für die Fourier-Transformation erhält man in Kurzschreibweise $QFT(|j\rangle) = F_{N\times N}\,|j\rangle$ für eine unitäre Matrix $F_{N\times N}$ mit den Elementen

$$F_{a,b} = \frac{1}{\sqrt{N}}\,(\omega_{1,N})^{ab} = \frac{1}{\sqrt{N}}\,e^{\frac{2\pi iab}{N}}.$$

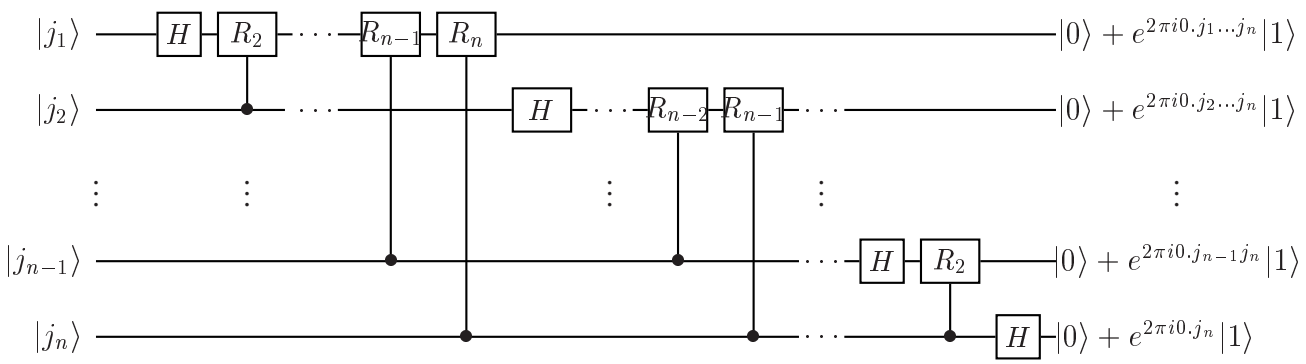
Wir legen nun fest, dass $N = 2^n$, so dass das Qubit $|j\rangle$ darstellbar ist als

$$|j\rangle = |\underbrace{0110...11}_{n}\rangle = |j_1, j_2, ..., j_n\rangle \text{ mit } j_i \in \{0, 1\}.$$

Außerdem legen wir eine Notation fest, in der die Zeichenfolge $.j_l j_{l+1}...j_n$, $j_i \in \{0, 1\}$ die Bedeutung $j_l\frac{1}{2} + j_{l+1}\frac{1}{4} + ... + j_n\frac{1}{2^{n-l+1}}$ erhält. Somit gilt z.B. $.011 = \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$.
Die Behauptung ist nun, dass sich die Quantum Fourier Transformation darstellen lässt als

$$\begin{aligned} QFT(|j\rangle) &= (|0\rangle + e^{2\pi i 0.j_n}|1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle)\cdots \\ &\quad \cdots(|0\rangle + e^{2\pi i 0.j_1...j_n}|1\rangle)\,\frac{1}{\sqrt{N}} \end{aligned} \tag{5.1}$$

Diese Darstellung wird auch als Produktdarstellung bezeichnet; aus ihr kann auf einfache Weise ein effizienter Quantenschaltkreis für die QFT abgeleitet werden:



Dabei symbolisiert R_k jeweils eine Matrix der Form

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

was eine unitäre Transformation darstellt. Die Hadamard-Transformation H ist bekanntlich als

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

definiert.
Wir wollen uns nun schrittweise klarmachen, dass man durch diesen Schaltkreis die oben angegebene Produktdarstellung der QFT erhält. Unser Anfangszustand sei $|j_1 ... j_n\rangle$. Durch die Hadamard-Transformation auf $|j_1\rangle$ erhält man den Zustand

$$H\,|j_1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1}|1\rangle\right)|j_2 ... j_n\rangle,$$

da $e^{2\pi i 0.j_1} = -1$ für $j_1 = 1$ und $e^{2\pi i 0.j_1} = +1$ für $j_1 = 0$.
Das Gatter R_2 wird nun kontrolliert in Abhängigkeit von $|j_2\rangle$ ausgeführt, d.h. man erhält als Folgezustände die beiden Alternativen

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1}|1\rangle\right)|j_2 ... j_n\rangle, \text{ falls } j_2 = 0$$
$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle\right)|j_2 ... j_n\rangle, \text{ falls } j_2 = 1$$

Aus dieser Fallunterscheidung wird allerdings ersichtlich, dass man den Folgezustand stets allgemein beschreiben kann durch die zweite Formulierung $\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle\right)|j_2 ... j_n\rangle$, da für $j_2 = 0$ die zweite Ziffer der Mantisse im Exponenten keinen Beitrag liefert. Nach Anwendung aller kontrollierten R_i-Gatter bis zum Gatter R_n erhält man in analoger Fortführung den Zustand

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 ... j_n}|1\rangle\right)|j_2 ... j_n\rangle.$$

Dies wird schließlich noch für die anderen Qubits ebenfalls durchgeführt und man erhält nach einer Vertauschung der Reihenfolge der Qubits, die mit einem einfachen weiteren Schaltkreis erzeugt werden kann, die gewünschte Produktdarstellung.
Insgesamt benötigt dieser Schaltkreis damit $\sum_{i=1}^{n} i = \frac{n(n-1)}{2} = O(n^2)$ Gatter, wobei es sich dabei um „kleine“, d.h. nicht besonders aufwendige Gatter handelt.

Wir versuchen nun, analog zur FFT eine rekursive Darstellung der QFT zu entwickeln um die Behauptung (5.1) zu beweisen. Die ursprüngliche Definition der QFT war

$$QFT(|j\rangle) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle$$

Wir definieren nun die Bitstrings $\hat{j} = j_2 ... j_n$ und $j' = j_1 \ldots j_{n-1}$. Außerdem schreiben wir die Darstellung der QFT um in

$$QFT(|j\rangle) = \frac{1}{\sqrt{N}} \sum_{k' \in \{0,1\}^{n-1}} e^{\frac{2\pi i j k' 0}{N}} |k'0\rangle + e^{\frac{2\pi i j k' 1}{N}} |k'1\rangle,$$

wobei $k'0$ und $k'1$ als konkatenierte Bitstrings zu verstehen sind. Mit den einfachen Überlegungen $bin(x0) = 2\,bin(x)$ und $bin(x1) = 2\,bin(x) + 1$ erhält man ferner

$$\begin{aligned} QFT(|j\rangle) &= \frac{1}{\sqrt{N}} \sum_{k' \in \{0,1\}^{n-1}} e^{\frac{2\pi i j k'}{N/2}} |k'0\rangle + e^{\frac{2\pi i j (2k'+1)}{N}} |k'1\rangle \\ &= \underbrace{\frac{1}{\sqrt{N}} \sum_{k' \in \{0,1\}^{n-1}} e^{\frac{2\pi i j k'}{N/2}}}_{=\sqrt{2}\, FT_{\frac{N}{2}}(|\hat{j}\rangle)} |k'\rangle \left(|0\rangle + e^{2\pi i j/N}\right) \end{aligned} \tag{5.2}$$

Die in der geschweiften Klammer behauptete Gleichheit gilt, da

$$e^{\frac{2\pi i j k'}{N/2}} = e^{\frac{2\pi i j_1 \hat{j} k'}{N/2}} = e^{\frac{2\pi i (2^{n-1} j_1 + \hat{j}) k'}{N/2}} = \underbrace{e^{\frac{2\pi i 2^{n-1} j_1 k'}{N/2}}}_{=1} e^{\frac{2\pi i \hat{j} k'}{N/2}} = e^{\frac{2\pi i \hat{j} k'}{N/2}}$$

Deshalb gilt mit Gleichung (5.2)

$$\frac{1}{\sqrt{N/2}} \sum_{k' \in \{0,1\}^{n-1}} e^{\frac{2\pi i \hat{j} k'}{N/2}} |k'\rangle \left(\frac{|0\rangle + e^{2\pi i \cdot j/N}}{\sqrt{2}}\right) = FT_{\frac{N}{2}} \left(|\hat{j}\rangle\right) \left(\frac{|0\rangle + e^{2\pi i \cdot j/N}}{\sqrt{2}}\right)$$

5.4 Der Algorithmus für die Faktorisierung

Die grundsätzliche Problemstellung der Faktorisierung ist – wie der Name schon sagt – für eine gegebene (meist sehr große) Zahl N, von der man weiß, dass es keine Primzahl ist, eine Zahl x zu finden, so dass $x \neq N$ und $x \neq 1$ mit $x|N$ (x Teiler von N). Die besten bekannten Algorithmen für dieses Problem, das vor allem bei Verschlüsselungsfragen eine Rolle spielt, liegen in der Komplexität exponentiell in der Länge der Binärdarstellung der Eingabe $|N|$, wobei $|N| = O(\log N)$.
Der Algorithmus von Shor löst nun nicht direkt das Faktorisierungs-Problem, sondern ein anderes Problem, auf das die Faktorisierung jedoch reduzierbar ist. In dem „Ersatzproblem“ geht es darum, bei gegebenem Y mit $0 < Y < N$, wobei $ggT(Y, N) = 1$, die Ordnung $Ord(Y) = min\{r \,|\, Y^r \equiv 1 \, mod \, N\}$ zu finden. In der Lösung zu diesem Problem ist der Rückgriff auf zwei Lemmata notwendig, die wir an dieser Stelle im voraus betrachten wollen.

Lemma 5.4.1 *Sei N nicht-prim. Falls $x^2 \equiv 1 \, mod \, N$ und $x \not\equiv \pm 1 \, mod \, N$, dann gilt*
$ggT(N, x+1) \neq 1, N$.

Beweis: Aus den Voraussetzungen folgt direkt, dass $x^2 - 1 \equiv 0 \, mod \, N$. Damit gilt $x^2 - 1 = (x-1)(x+1) = kN$ für ein $k \in \mathbb{N}$. Laut Vorraussetzungen teilt N jedoch $(x-1)$ und $(x+1)$ jeweils nicht. Wenn N nun aber wie gesehen das Produkt $(x-1)\,(x+1)$ teilt, so muss ein nicht-trivialer Faktor von N existieren, der $(x+1)$ teilt. □

Lemma 5.4.2 *Sei $N = p_1^{e_1} \cdots p_m^{e_m}$, p_i prim und $p_i \neq 2 \, \forall i$, d.h. N ungerade. Sei ferner r_y die Ordnung von y in $\mathbb{Z}_N$. Dann ist die Wahrscheinlichkeit*

$$Prob(r_y \textit{ gerade und } y^{r_y/2} \not\equiv \pm 1 \, mod \, N, \; y \in \mathbb{Z}_N) \geq 1 - 2^{-m} \geq \tfrac{1}{2}.$$

Auf den Beweis dieses Lemmas wird an dieser Stelle verzichtet. Ein Beweis kann z.B. im Buch von Nielsen und Chuang nachgelesen werden.

Faktorisierung von N: Auf Basis der allgemeinen Problembetrachtung und der Lemmata können wir nun ein Schema zur Faktorisierung betrachten, wobei wir annehmen, dass unsere Eingabe N ungerade und teilbar ist:

```
Wähle  y ∈R ℤ
fertig, falls y Teiler von N; ansonsten:
Finde  r = ord_N(y)
falls r gerade und y^{r/2} ≢ ±1 mod N  do
   x:=y^{r/2}
   finde ggT(x+1,N)
Ausgabe ggt(x+1,N)
```

Da nach dem Lemma die Bedingung in der vierten Zeile nicht immer erfüllt ist, kann der Algorithmus zur Reduktion der Fehlerwahrscheinlichkeit k-mal wiederholt werden.

5.5 Der Algorithmus zum Ordnungsproblem

Die eigentliche Leistung von Shor liegt darin, den im folgenden dargestellten Algorithmus zur Lösung des Ordnungsproblems entwickelt zu haben, der einen wesentlichen Teil des Faktorisierungs-Algorithmus ausmacht und auch den Quantenanteil ins Spiel bringt. Die Eingaben des Algorithmus sind Y und N. Q ist eine Zahl größer als N (z.B. $Q = N^2$). Wir notieren die Längen von N und Q als n und q. Die einzelnen Schritte des Algorithmus sind:

Eingabe (Y, N), **Finde** $\min\{r \mid Y^r \equiv 1 \, mod \, N\}$

Voraussetzung: Register $|0^q\rangle|0^n\rangle$

(1) Fourier-Transformation auf dem ersten Register

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

(2) Berechne die modulare Potenzfunktion $Y^k \mathrm{mod} N$ auf dem zweiten Register

$\frac{1}{\sqrt{Q}} \sum_{k=0}^{q-1} |k\rangle \otimes |Y^k \mathrm{mod} N\rangle$

(3) Messung auf dem zweiten Register

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

(4) Fourier-Transformation auf dem ersten Register

$\frac{1}{\sqrt{Q}} \sum_{l=0}^{q-1} (\frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} e^{2\pi i (jr+k_0) l/Q}) |l\rangle \otimes |Y^{k_0} \mathrm{mod} N\rangle$

(5) Messung auf dem ersten Register

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

Wir wollen diese Schritte nun im einzelnen betrachten und näher erläutern.
Der Algorithmus arbeitet auf zwei Registern, wobei das erste q Elemente $(0 \ldots q-1)$ und das zweite n Elemente $(0 \ldots n-1)$ enthält. In Punkt (1) wird einfach eine Quantum Fourier Transformation auf dem ersten Register angewendet, was dasselbe Ergebnis liefert wie eine Hadamard-Transformation. In Schritt (2) wird die modulare Potenzfunktion auf den hinteren n Qubits berechnet, von der bekannt ist, dass es effiziente Lösungen gibt (siehe Übungsaufgabe 13). Interessant wird es nun bei der Messung des zweiten Registers in Schritt (3): Wenn r die Ordnung von Y sein soll, gibt es bei dieser Messung höchstens r unterschiedliche Werte von $Y^k \, mod \, N$, da für Exponenten $k = j\,r + k_0$ mit $j \geq 1$ und $0 \leq k_0 < r$ stets gilt

$$\begin{aligned} Y^k \, mod \, N &= Y^{jr+k_0} \, mod \, N = (Y^r)^j \, Y^{k_0} \, mod \, N \\ &= (\underbrace{Y^r \, mod \, N}_{=1})^j \, (Y^{k_0} \, mod \, N) = Y^{k_0} \, mod \, N \end{aligned}$$

Deshalb können wir mit $A = \lfloor \frac{Q}{r} \rfloor$ das Messergebnis wie oben bereits angewendet in vereinfachter Form schreiben:

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

In Schritt (4) wird nochmals eine QFT auf das erste Register angewendet, was dann schließlich dazu führt, dass in Schritt (5) ein Qubit $|l\rangle$ gemessen werden kann. Die oben angegebene Wahrscheinlichkeit kann mit der Regel $|a\,b| = |a|\,|b|$ noch vereinfacht werden zu

$$\begin{aligned} Prob(|l\rangle) &= |\frac{1}{\sqrt{QA}} \sum_{j=0}^{A-1} e^{2\pi i(jr+k_0)l/Q}|^2 \\ &= \frac{1}{QA}\,|e^{2\pi i k_0 l/Q} \sum_{j=0}^{A-1} (e^{2\pi i r l/Q})^j|^2 \\ &= \frac{1}{QA}\,|\sum_{j=0}^{A-1} (e^{2\pi i r l/Q})^j|^2 \end{aligned}$$

Mit diesem Verfahren erhält man als Ergebnis der Messung also einen Wert $l \in 0 \ldots Q-1$, mit dessen Hilfe nun die Ordnung von Y bezüglich N mit einer gewissen Fehlerwahrscheinlichkeit ermittelt werden kann. Dabei gibt es zwei verschiedene Fälle, in denen die Methodik jeweils etwas unterschiedlich ist.

Fall 1: r teilt Q, d.h. $A = \frac{Q}{r}$
In diesem (einfacheren) Fall gilt für die Summe

$$\sum_{j=0}^{A-1} (e^{2\pi i r l/Q})^j = \begin{cases} A & \text{falls } e^{2\pi i r l/Q} = 1 \\ \frac{e^{2\pi i r l A/Q}-1}{e^{2\pi i r l/Q}-1} = 0 & \text{sonst} \end{cases}$$

Damit gilt für die Wahrscheinlichkeit bei der Messung

$$Prob(Messung = l) = \begin{cases} \frac{1}{QA}\,A^2 = \frac{1}{r} & \text{falls } e^{2\pi i r l/Q} = 1 \\ 0 & \text{sonst} \end{cases}$$

Es wird also mit Wahrscheinlichkeit 1 ein Wert gemessen, der der Bedingung $e^{2\pi i r l/Q} = 1$ genügt, d.h. es muss gelten $lr = mQ \Leftrightarrow \frac{l}{Q} = \frac{m}{r}$ für ein $m \in \mathbb{N}$. Da $l \in 0 \ldots Q-1$ ist, gilt, dass $lr \in 0 \ldots (Q-1)r$, so dass sich wiederum mit $lr = mQ$ ergibt, dass $m \in 0 \ldots r-1$ (das m wird dabei durch die Messung zufällig unter Gleichverteilung aus den r Werten gewählt).
Wir kennen also die Werte Q und nach der Messung auch l und wollen damit nun r bestimmen. Mit dem Zusammenhang $\frac{l}{Q} = \frac{m}{r}$ gilt für den Fall, dass m und r teilerfremd sind, d.h. $ggT(m,r) = 1$, dass r gleich dem Nenner des gekürzten Bruches $\frac{l}{Q}$ ist, also formal:

$$\text{Falls } ggT(m,r) = 1 \Rightarrow r = \frac{Q}{ggT(l,Q)}$$

Wie können wir nun aber wissen, ob m und r teilerfremd sind? Um sicherzugehen, fordern wir, dass m eine Primzahl sein soll, deren größter gemeinsamer Teiler mit jeder anderen Zahl 1 ist. Man kann zeigen, dass es $\geq \frac{r}{\log r}$ Primzahlen gibt, die kleiner als r sind. Somit können wir eine Fehlerwahrscheinlichkeit angeben, die meist eine recht grobe Abschätzung liefert:

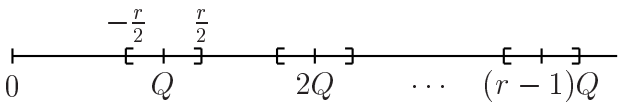
$$Prob(ggT(m,r) = 1) \geq Prob(m \text{ Primzahl}) \geq \frac{\frac{r}{\log r}}{r} = \frac{1}{\log r} \overset{r<N}{\geq} \frac{1}{\log N}$$

Wenn wir den Algorithmus t-mal wiederholen, wobei $t = k \log N$, so erhalten wir jeweils t Werte für l und damit für r, aus denen wir das kleinste r mit $Y^r \equiv 1 \, mod \, N$ auswählen. Insgesamt erhalten wir so eine Fehlerwahrscheinlichkeit von

$$Prob(Fehler) = (1 - \tfrac{1}{\log N})^t = (1 - \tfrac{1}{\log N})^{k \log N} \leq e^{-k},$$

d.h. man erreicht mit einer linearen Erhöhung der Anzahl von Wiederholungen eine exponentielle Verringerung der Fehlerwahrscheinlichkeit.

Fall 2: $A = \lfloor \frac{Q}{r} \rfloor < \frac{Q}{r}$
In diesem Fall vereinfacht sich die Summe in dem Ausdruck für die Wahrscheinlichkeit von l nicht, da der Exponentialterm auf jeden Fall eine nicht-triviale Einheitswurzel darstellt. Man kann nun allerdings zwei verschiedene Fälle klassifizieren, nämlich den Fall, dass $lr \, mod \, Q$ einen hohen Wert liefert, und den Fall, dass $lr \, mod \, Q \approx 0$. Wir beschränken unsere Betrachtung auf den zweiten Fall und konkretisieren die Bedingung, indem wir fordern, dass $-\frac{r}{2} < lr \, mod \, Q \leq \frac{r}{2}$. Wegen $l \in 0 \ldots Q-1$ gilt $lr = 0 \ldots (Q-1)r$; diese Menge enthält r Vielfache von Q, was man sich folgendermaßen veranschaulichen kann:



Da die Intervalle jeweils die Länge r haben, wird klar, dass pro Intervall immer nur ein Wert von lr existieren kann, der darin liegt ($(l-1)r$ bzw. $(l+1)r$ sind bereits um r kleiner bzw. größer). Somit gibt es genau r Werte l, die die Eigenschaft $-\frac{r}{2} < lr \, mod \, Q \leq \frac{r}{2}$ erfüllen.
Für solche Werte gilt wegen der obigen Überlegung

$$e^{2\pi i lr/Q} = e^{2\pi i(nQ+m)/Q} = e^{2\pi i m/Q}, \text{ wobei } -\tfrac{r}{2} < m \leq \tfrac{r}{2}$$

Somit ist die Wahrscheinlichkeit

$$\begin{aligned} Prob(Messung = l) &= \frac{1}{QA} |\sum_{j=0}^{A-1} (e^{2\pi i m/Q})^j|^2 \\ &\geq \frac{1}{QA} |\sum_{j=0}^{A-1} (e^{2\pi i \frac{r}{2}/Q})^j|^2 \\ &= \frac{1}{QA} |\frac{1 - e^{\pi i r A/Q}}{1 - e^{\pi i r/Q}}|^2 \\ &= \frac{1}{QA} \frac{(1 - \cos(\pi r A/Q))^2 + \sin^2(\pi r A/Q)}{(1 - \cos(\pi r/Q))^2 + \sin^2(\pi r/Q)} \\ &= \frac{1}{QA} \frac{2 - 2\cos(\pi r A/Q)}{2 - 2\cos(\pi r/Q)} \end{aligned}$$

Dies lässt sich schließlich mit Hilfe der Formel für den halben Winkel $\sin\frac{\theta}{2} = \pm\sqrt{\frac{1}{2}(1-\cos\theta)}$ umschreiben zu

$$Prob(Messung = l) = \frac{1}{QA} \frac{\sin^2 \frac{\pi r A}{2Q}}{\sin^2 \frac{\pi r}{2Q}}$$

Mit $A = \lfloor \frac{Q}{r} \rfloor$ gilt $\frac{\pi r A}{2Q} \approx \frac{\pi}{2}$. Ebenso gilt wegen $Q \gg N > r$, dass $\frac{\pi r}{2Q}$ sehr klein ist und somit die Näherung $\sin\frac{\pi r}{2Q} \approx \frac{\pi r}{2Q}$ verwendet werden kann. Insgesamt erhalten wir so die Näherung

$$Prob(Messung = l) = \frac{1}{QA} \frac{\sin^2 \frac{\pi r A}{2Q}}{\sin^2 \frac{\pi r}{2Q}} \approx \frac{1}{QA} \frac{1}{\frac{\pi^2 r^2}{4Q^2}} \approx \frac{4}{\pi^2 r}$$

Da die Näherung den Wert kleiner macht und es wie bereits gesehen r Möglichkeiten für ein l mit der gewünschten Eigenschaft gibt, gilt

$$Prob(\text{ein l mit } -\tfrac{r}{2} < lr \bmod Q \leq \tfrac{r}{2} \text{ wird gemessen}) \geq \tfrac{4}{\pi^2}$$

Wenn wir nun ein solches l mit mindestens dieser Wahrscheinlichkeit erhalten, so wissen wir, dass es ein $m \in \mathbb{N}$ gibt mit $|lr - mQ| \leq \frac{r}{2}$. Daraus folgt durch Umformung, dass

$$|\frac{l}{Q} - \frac{m}{r}| \leq \frac{1}{2Q}$$

Um den Wert von r zu erhalten, hilft uns nun ein Lemma weiter, das wir jedoch nicht beweisen wollen.

Lemma 5.5.1 *Es gibt höchstens ein Paar* (a, b) *mit* $|\frac{l}{Q} - \frac{a}{b}| \leq \frac{1}{2Q}$ *und* $b < N, Q > N^2$.

Zur weiteren Vorgehensweise wird die Methode der Kettenbrüche („continued fractions“) benötigt, die wir im folgenden abstrakt und an einem Beispiel kurz einführen wollen.
Gegeben sei eine Zahl $\alpha \in \mathbb{R}$. Gesucht sind Zahlen $a_0, a_1, ..., a_n, ... \in \mathbb{Z}$, so dass

$$\alpha = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + ...}}}$$

Formal erhält man die Werte a_i auf folgende Weise:
Logischerweise gilt $a_0 = \lfloor \alpha \rfloor$ und damit $\alpha = a_0 + x_0$ mit $x_0 = \alpha - a_0 = \alpha - \lfloor \alpha \rfloor$. Dies können wir wieder umschreiben als $\alpha = a_0 + \frac{1}{\frac{1}{x_0}} = a_0 + \frac{1}{\frac{1}{\lfloor x_0 \rfloor} + x_1}$ mit $x_1 = \frac{1}{x_0} - \frac{1}{\lfloor x_0 \rfloor}$, also gilt $a_1 = \frac{1}{\lfloor x_0 \rfloor}$, usw.
Wenn wir davon ausgehen, dass der Kettenbruch nach n Schritten terminiert, so lässt sich dieses Schema in analoger Weise fortführen, bis man für $a_{n+1} = 0$ erhält.
Als Beispiel wollen wir die Zahl $\alpha = \frac{55}{17}$ betrachten. Das Schema liefert

$$\alpha = \frac{55}{17} = 3 + \frac{4}{17} = 3 + \frac{1}{\frac{17}{4}} = 3 + \frac{1}{4 + \frac{1}{4}}$$

Oft beschränkt man sich auch absichtlich auf eine bestimmte Anzahl von Koeffizienten a_i. Mit der Funktion

$$CF_n(\alpha) = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_n}}} = \frac{p_n}{q_n}, p_n, q_n \in \mathbb{Z}$$

erreicht man somit für beliebige reelle Zahlen eine rationale Näherung, zu deren Eigenschaften das folgende Lemma zwei Aussagen macht.

Lemma 5.5.2 *(i) $CF_n(\alpha) = \frac{p_n}{q_n}$ ist die beste Approximation von α mit Nenner $\leq q_n$. (ii) Falls $\alpha \in \mathbb{Q}$, findet die Methode in höchstens $O(n)$ Etappen das Ergebnis, wobei n die Anzahl der Bits von p und q sind. In diesem Fall gilt $\alpha = \frac{p}{q}$.*

Kehren wir nun zurück zu unserem eigentlichen Anliegen: Wir approximieren mit der Methode der Kettenbrüche den Quotienten $\frac{l}{Q} = \frac{p_n}{q_n}$, bis $q_n > N$ und $|\frac{l}{Q} - \frac{p_n}{q_n}| \leq \frac{1}{2Q}$. Damit folgt aus dem oberen Lemma, dass $\frac{p_m}{q_n} = \frac{m}{r}$, woraus wir, wenn m und r teilerfremd sind, einen Wert für r erhalten (die Brüche liegen beide in gekürzter Form vor). Die Wahrscheinlichkeit, dass m und r teilerfremd sind kann wie im Fall 1 der Analyse abgeschätzt werden.
Da wir jedoch wieder nur mit einer gewissen Wahrscheinlichkeit ein geeignetes l messen, wiederholen wir auch diesen Teil des Algorithmus t-mal und wählen aus den erhaltenen Werten für r die minimale Zahl aus, die die Eigenschaft $Y^r \equiv 1 \, mod \, N$ erfüllt.

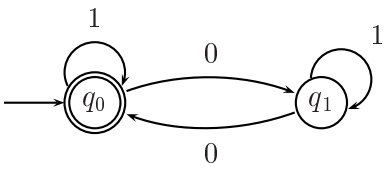
Kapitel 6

Endliche Quanten-Automaten

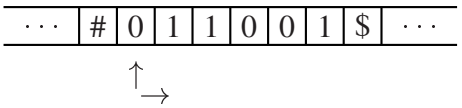
Ein großes Problem in der praktischen Umsetzung von Quanten-Schaltkreisen oder „Quantencomputern“ besteht zur Zeit noch in der Bereitstellung einer ausreichenden Menge von Qubits. Wie wir im Algorithmus von Shor gesehen haben, werden Register mit einer Vielzahl von Qubits benötigt, die von der Länge der zu faktorisierenden Zahl abhängt; bisherige praktische Lösungen von Qubit-Registern bewegen sich im Rahmen einstelliger Werte für die Anzahl.
Eine interessante Möglichkeit, mit wenigen Qubits bereits verwertbare Ergebnisse zu erzielen, stellen endliche Quanten-Automaten dar, die auf Basis der klassischen Automaten definiert werden. Im folgenden werden wir uns zur Gewöhnung an die Notation zunächst mit dem 1QFA beschäftigen, der allerdings noch nicht mächtiger ist als klassische Varianten.

6.1 Der One Way Quantum Finite Automaton (1QFA)

Der klassische endliche Automat (DFA) ist definiert als ein Tupel $A = (Q, \Sigma, \delta, q_0, Q_E)$ wobei $\delta : Q \times \Sigma \to Q$ die Überführungsfunktion ist (vergleiche auch Theoretische Informatik II). Schematisch werden DFAs meist folgendermaßen dargestellt:



Solche endlichen Automaten können auch als Turing-Maschine betrachtet werden, die nur lesen kann. Im Modell der Turing-Maschine bedeutet das, dass der Schreib-Lese-Kopf lediglich eine Bewegung nach rechts machen darf und damit nie eine bereits verlassene Position nochmals erreicht, weshalb jede Schreiboperation ins Leere läuft. Schematisch lässt sich ein Automat der besprochenen Art also auch auf folgende Weise verdeutlichen:



Das Eingabewort wird sequentiell gelesen und nur akzeptiert, wenn jeder Schritt nach rechts einer Regel der Überführungsfunktion entspricht. An dieser Grafik wird deutlich, weshalb man auch oft von „One Way DFAs“ spricht.
Ausgehend von diesem Modell kann man sich nun die Frage stellen, was passieren würde, wenn die Überführungsfunktion den Gang in die andere Richtung oder das Stehenbleiben ebenfalls zulassen würde, also von der Gestalt $\delta : Q \times \Sigma \to Q \times \{L, R, N\}$ wäre. Tatsächlich gibt es diesen Ansatz auch unter dem Namen „Two Way DFA“ bzw. 2DFA, allerdings kann man recht einfach zeigen, dass dieses Modell nicht mächtiger ist als 1DFAs.

Definition 6.1.1 (1QFA) *Ein 1QFA wird festgelegt durch ein Tupel* $A = \{\Sigma, Q, q_0, Q_a, Q_r, \delta\}$*, wobei*

- Σ *das Eingabealphabet*
- Q *die Menge aller Zustände*
- q_0 *der Startzustand*

- $Q_a \subseteq Q$ *die Menge der akzeptierenden Zustände und* $Q_r \subseteq Q$ *die Menge der verwerfenden Zustände (*r *für „reject“); für diese gilt* $Q_a \cap Q_r = \emptyset$
- $\delta : Q \times \Gamma \times Q \to \mathbb{C}$ *die Überführungsfunktion*
- $\Gamma = \Sigma \cup \{\#, \$\}$ *das Arbeitsalphabet.*

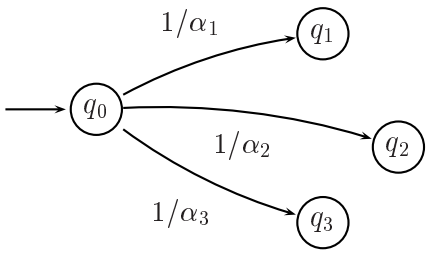
Wir bezeichnen ferner die Menge $\{|q\rangle \mid q \in Q\}$ als Menge der *Basiszustände*. Der komplexe Koeffizient, der sich aus der Überführungsfunktion ergibt, z.B. $\delta(q_i, b, q_j) = \alpha$, stellt im Folgezustand von q_i die Amplitude des Zustands q_j dar. Abstrakt kann man dies auch für alle $b \in \Gamma$ durch einen Linearoperator V_b formulieren, für den gilt

$$V_b(|q\rangle) = \sum_{q' \in Q} \delta(q, b, q')|q'\rangle, \text{ wobei } \sum_{q' \in Q} |\delta(q, b, q')|^2 = 1$$

Die Überführungsfunktion δ erfüllt auch die Orthogonalitätseigenschaft:

$$\sum_{q'} \delta(q_1, b, q')\delta(q_2, b, q') = \begin{cases} 1 & \text{falls } q_1 = q_2 \\ 0 & \text{sonst} \end{cases}$$

Anschaulich wird dies an einem einfachen Beispiel klar (die Pfeile sind dabei mit dem zu lesenden Zeichen und nach einem Schrägstrich mit der Wahrscheinlichkeitsamplitude für den jeweiligen Übergang beschriftet):



In dieser Grafik ist der Übergang $|q_0\rangle \xrightarrow{1} \alpha_1 |q_1\rangle + \alpha_2 |q_2\rangle + \alpha_3 |q_3\rangle$ dargestellt; als Nebenbedingung gilt $\sum_i |\alpha_i|^2 = 1$.
Alle Zustände, in denen der 1QFA anhält (HALT-Zustände), sind durch Q_a und Q_r festgelegt. Bei den restlichen Zuständen $Q_n = Q \setminus (Q_a \cup Q_r)$ werden jeweils Messungen vorgenommen. Damit können wir die Gesamtheit aller möglichen Ergebnisse eines Automaten darstellen durch das Tensorprodukt $E_a \otimes E_r \otimes E_n$ mit $E_a = span\{|q\rangle \mid q \in Q_a\}$, $E_r = span\{|q\rangle \mid q \in Q_r\}$ und $E_n = span\{|q\rangle \mid q \in Q_n\}$.

Jede Messung $|\psi\rangle$ stellt somit eine Projektion von $|\psi\rangle$ auf einen Zustand $|\psi_i\rangle$ in E_a, E_r oder E_n mit Wahrscheinlichkeit $\||\psi_i\rangle\|^2$ dar.
Wenn wir zum Beispiel festlegen, dass $Q_a = \{q_0\}$, $Q_r = \{q_1\}$ und $Q_n = \{q_2, q_3\}$, so ergibt sich für einen Zustand $\alpha_0 |q_0\rangle + \alpha_1 |q_1\rangle + \alpha_2 |q_2\rangle + \alpha_3 |q_3\rangle$, dass man bei einer Messung mit Wahrscheinlichkeit $|\alpha_0|^2$ den einzigen Zustand aus Q_a, den Zustand $|q_0\rangle$, erhalten würde. Analog ergibt sich für $|q_1\rangle$, den einzigen Zustand aus Q_r, eine Wahrscheinlichkeit von $|\alpha_1|^2$. Aus Q_n gibt es nun mehrere mögliche Zustände, weshalb hier die Linearkombination erhalten bleibt. Für die Wahrscheinlichkeit ergibt sich deshalb $|\alpha_2|^2 + |\alpha_3|^2$, und der Folgezustand ist nach Normierung $\sqrt{\frac{1}{|\alpha_2|^2+|\alpha_3|^2}}(\alpha_2 |q_2\rangle + \alpha_3 |q_3\rangle)$.

Für eine Eingabe $\#b_1 \dots b_n\$$ ergibt sich mit diesem Schema:

- Anwendung des Linearoperators $V_\#$: $V_\# |q_0\rangle \to |\psi\rangle$
- Messung ergibt $|\psi'\rangle$
- Falls $|\psi'\rangle \in E_a$: HALT und akzeptieren
  Falls $|\psi'\rangle \in E_r$: HALT und verwerfen
- sonst: $|\psi'\rangle \in E_n$, Normierung zu $|\psi''\rangle$
- Fortfahren mit Linearoperator V_{b_1} auf den normierten Zustand: $V_{b_1} |\psi''\rangle$

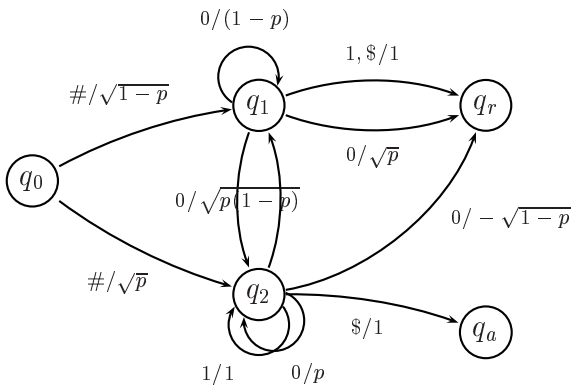
Schließlich erhält man damit einen Zustand $P_n V_{b_n} \dots V_{b_2} P_n V_{b_1} P_n V_\# |q_0\rangle$, wobei P_n die Projektion in E_n symbolisiert.

Die Frage ist nun, wie man mit solchermaßen definierten Übergängen und Abbruchmöglichkeiten eine Sprache definieren kann. Dazu definieren wir uns zu einem 1QFA A ein Tripel $(\psi, p_a, p_r) \in V_A = Q \times \mathbb{R} \times \mathbb{R}$, das eine erweiterte Beschreibung des Zustands liefert: A befindet sich zu einem gewissen Zeitpunkt in (ψ, p_a, p_r), falls A die bis jetzt vorliegende Eingabe mit Wahrscheinlichkeit p_a akzeptiert, sie mit Wahrscheinlichkeit p_r verwirft und mit Wahrscheinlichkeit $1 - p_a - p_r = \|\psi\|^2$ weitermacht (ψ ist in diesem Falle nicht normiert). Für einen Übergang $T_b : V_A \to V_A$ ergibt sich $(\psi, p_a, p_r) \to (P_n V_b \psi, p_a + \|P_a V_b \psi\|^2, p_r + \|P_r V_b \psi\|^2)$. Für ein Eingabewort x gilt somit $T_{\#x\$} = T_\$ T_{b_n} \dots T_{b_1} T_\# (|q_0\rangle, 0, 0)$.
Nach Ermittlung dieses Ausdrucks können wir also die Wahrscheinlichkeiten angeben, mit denen A das Wort x akzeptiert oder verwirft. Damit können wir den Akzeptanzbegriff für 1QFAs festlegen.

Definition 6.1.2 *Ein 1QFA A akzeptiert die Sprache $L \subseteq \Sigma^*$ mit Wahrscheinlichkeit $\frac{1}{2} + \epsilon$, falls*

1. $x \in L \Rightarrow Prob(A \textit{ akzeptiert } x) \geq \frac{1}{2} + \epsilon$
2. $x \notin L \Rightarrow Prob(A \textit{ verwirft } x) \geq \frac{1}{2} + \epsilon$

Als Beispiel wollen wir uns nun die Sprache $L = \{0^* 1^*\}$ anschauen. Wir definieren einen 1QFA A mit $Q = \{q_0, q_1, q_2, q_a, q_r\}$, $Q_a = \{q_a\}$, $Q_r = \{q_r\}$. Die restlichen Eigenschaften legen wir durch folgendes Schaubild von A fest:



Aus dieser grafischen Darstellung kann man folgende Übergänge ablesen:

$$\begin{aligned}
V_{\#} |q_0\rangle &= \sqrt{1-p}\,|q_1\rangle + \sqrt{p}\,|q_2\rangle \\
V_0 |q_1\rangle &= (1-p)\,|q_1\rangle + \sqrt{p(1-p)}\,|q_2\rangle + \sqrt{p}\,|q_r\rangle \\
V_0 |q_2\rangle &= \sqrt{p(1-p)}\,|q_1\rangle + p\,|q_2\rangle - \sqrt{1-p}\,|q_r\rangle \\
V_1 |q_1\rangle &= |q_r\rangle, V_1 |q_2\rangle = |q_2\rangle \\
V_\$ |q_1\rangle &= |q_r\rangle, V_\$ |q_2\rangle = |q_a\rangle
\end{aligned}$$

Wir wollen nun analysieren, wie sich der definierte Automat bei einer Eingabe x bzw $\#x\$$ verhält. Dazu unterscheiden wir drei Fälle, die bezüglich der vorgegebenen Sprache charakteristisch sind.

Fall 1: $x = 0^i$

$$\begin{aligned}
|q_0\rangle &\xrightarrow{\#} \sqrt{1-p}\,|q_1\rangle + \sqrt{p}\,|q_2\rangle \\
&\xrightarrow{0} \sqrt{1-p}\,\Big((1-p)\,|q_1\rangle + \sqrt{p(1-p)}\,|q_2\rangle + \sqrt{p}\,|q_r\rangle\Big) \\
&\quad + \sqrt{p}\,\Big(\sqrt{p(1-p)}\,|q_1\rangle + p\,|q_2\rangle - \sqrt{1-p}\,|q_r\rangle\Big) \\
&= \sqrt{1-p}\,|q_1\rangle + \sqrt{p}\,|q_2\rangle \\
&= \ldots\ldots \\
&\xrightarrow{\$} \sqrt{1-p}\,|q_r\rangle + \sqrt{p}\,|q_a\rangle
\end{aligned}$$

Die Analyse wird dadurch vereinfacht, dass das Lesen einer 0 den Zustand nicht verändert und somit ohne Zustandsänderung der Bitstring 0^k gelesen wird. Insgesamt wird das Wort also mit Wahrscheinlichkeit p akzeptiert und mit Wahrscheinlichkeit $1-p$ verworfen.

Fall 2: $x = 0^i 1^j$

$$\begin{aligned} |q_0\rangle &\overset{\#}{\to} \sqrt{1-p}\,|q_1\rangle + \sqrt{p}\,|q_2\rangle \\ &\overset{0}{\to} \ldots\ldots \\ &\overset{1}{\to} \sqrt{1-p}\,|q_r\rangle + \sqrt{p}\,|q_2\rangle \end{aligned}$$

An dieser Stelle wird die Eingabe mit Wahrscheinlichkeit $1-p$ verworfen, mit Wahrscheinlichkeit p gelangt man jedoch in Zustand $|q_2\rangle$ und man erhält des weiteren:

$$|q_2\rangle \overset{1^{j-1}}{\to} |q_2\rangle \overset{\$}{\to} |q_a\rangle$$

Insgesamt wird das Wort also auch hier mit einer Wahrscheinlichkeit von p akzeptiert.

Fall 3: $x \notin 0^*1^*$ Alle Wörter x, die nicht in der Sprache enthalten sind, sind von der Form $x = 0^i 1^j 0^k w$, wobei $i \geq 0$ und $k, j > 0$. Damit erhält man

$$\begin{aligned} |q_0\rangle &\overset{\#0^i}{\to} \sqrt{1-p}\,|q_1\rangle + \sqrt{p}\,|q_2\rangle \\ &\overset{1}{\to} \sqrt{1-p}\,|q_r\rangle + \sqrt{p}\,|q_2\rangle \end{aligned}$$

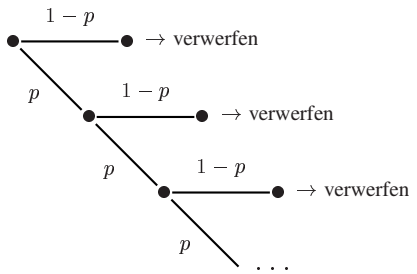
Somit gelangt der Automat bereits an dieser Stelle mit einer Wahrscheinlichkeit von $1-p$ in einen verwerfenden HALT-Zustand; mit Wahrscheinlichkeit p fährt er jedoch fort mit

$$|q_2\rangle \overset{1^{j-1}}{\to} |q_2\rangle \overset{0}{\to} \sqrt{p(1-p)}\,|q_1\rangle + p\,|q_2\rangle - \sqrt{1-p}\,|q_r\rangle$$

An dieser Stelle gelangt der Automat erneut mit Wahrscheinlichkeit $1-p$ in einen verwerfenden Zustand. Mit Wahrscheinlichkeit p hingegen gelangt er in den Zustand $\frac{\sqrt{p(1-p)}}{\sqrt{p}}\,|q_1\rangle + \frac{p}{\sqrt{p}}\,|q_2\rangle = \sqrt{1-p}\,|q_1\rangle + \sqrt{p}\,|q_2\rangle$ (wie bereits erwähnt, ist eine Normierung erforderlich). Damit ist der Automat wieder in einem Zustand angelangt, in dem er beliebig viele 0en ohne effektive Zustandsänderung lesen kann. Aus diesem Zustand gibt es also nur zwei mögliche Ausgänge: entweder wird eine 1 oder das Abschlusszeichen $\$$ gelesen. Für diese beiden Fälle gilt

$$\begin{aligned} \sqrt{1-p}\,|q_1\rangle + \sqrt{p}\,|q_2\rangle &\overset{1}{\to} \sqrt{1-p}\,|q_r\rangle + \sqrt{p}\,|q_2\rangle \\ \sqrt{1-p}\,|q_1\rangle + \sqrt{p}\,|q_2\rangle &\overset{\$}{\to} \sqrt{1-p}\,|q_r\rangle + \sqrt{p}\,|q_a\rangle \end{aligned}$$

Für die erste Variante wiederholt sich der Vorgang beliebig oft, wobei jeweils mit einer Wahrscheinlichkeit von $1 - p$ verworfen wird. Die Wahrscheinlichkeit, dass der Automat weitermacht, liegt jeweils bei p. Dies lässt sich auch schematisch verdeutlichen:



Wenn wir uns auf die ersten beiden Zweige beschränken, erhalten wir eine Abschätzung für die Gesamtwahrscheinlichkeit, mit der die Eingabe verworfen wird:

$$Prob(x \text{ wird verworfen}) \geq (1-p) + p(1-p) + p^2(1-p) = 1 - p^3$$

Damit gilt zusammengefasst

$$\begin{aligned} x \in L &\Rightarrow Prob_{Akz} \geq p \\ x \notin L &\Rightarrow Prob_{Verw} \geq 1 - p^3 \end{aligned}$$

Die optimale Wahl für p erhält man durch Gleichsetzen dieser beiden Wahrscheinlichkeiten. Die numerische Lösung der Gleichung $p = 1 - p^3$ ergibt einen Wert von $p \approx 0.68$.

6.2 Der Two-Way Quantum Finite Automaton (2QFA)

Das Modell des 1QFA ist in seiner Mächtigkeit beschränkt, da z.B. die Sprache $(0+1)^*0 \notin 1QFA$ ist. In der Tat lässt sich zeigen, dass gilt $L(1QFA) \subset REG$. Aus diesem Grunde ist es sinnvoll, das Modell zu einem 2QFA zu erweitern, in dem die Richtung der Abarbeitung festgelegt werden kann. Schauen wir uns jedoch zunächst eine analoge Erweiterung des DFAs an, den 2DFA. Die Änderung der Definition wirkt sich nur in der Übergangsfunktion aus, sie ist nun definiert als $\delta : Q \times \Gamma \to Q \times \{L, R, N\}$, d.h. mit jedem Schritt wird auch festgelegt, in welche Richtung der Automat weiterläuft. Allerdings ist dieses Modell nicht mächtiger als ein normaler DFA.

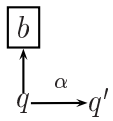
Zur Einführung des 2QFAs wollen wir erneut einen Eingabestring der Form $\#b_1 \dots b_n\$$ betrachten. Unter einer Konfiguration verstehen wir die Kombination aus einem Zustand und einer Position im Eingabestring. Bei einer Eingabe der Länge n gibt es also $n\,|Q|$ mögliche Konfigurationen. Damit lassen sich Quantumzustände definieren mit $|i, q\rangle$, $q \in Q$ und $i \in 1 \dots n$. Die Gesamtheit dieser Zustände bezeichnen wir mit C_n. Damit können wir jeden Zustand als Superposition

$$|\psi\rangle = \sum_{c \in C_n} \alpha_c \, |c\rangle$$

beschreiben. In dem Modell erfolgt nach jeder Bewegung der Position eine Messung.
Die Übergangsfunktion wird nun definiert als

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

wobei $-1 = L$ einen Schritt in der Eingabe nach links, $0 = N$ keinen Schritt und $1 = R$ einen Schritt nach rechts symbolisiert. Die komplexe Zahl, die den Wert der Überführungsfunktion ausmacht, stellt die Amplitude für den Folgezustand dar. Schematisch lässt sich dies so darstellen:



In diesem Schema hat der neue Zustand, der sich bei Beibehalten der Position in der Eingabe und Ändern des Zustands zu q' ergibt, die Amplitude α.
Betrachten wir nun wieder eine konkrete Eingabe $x = \#x_1 \dots x_n\$$. Angewendet auf diese Eingabe können wir alle relevanten Beiträge der Überführungsfunktion zusammengefasst in einer Transformation U_δ^x definieren mit

$$U_\delta^x |k, q\rangle = \sum_{q', d} \delta(q, x_k, q', d) \, |(k + d) mod |x|, q'\rangle$$

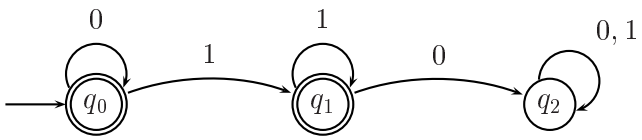
Dabei stellt der Modulo-Term nur sicher, dass man mit der Position im Eingabestring nicht über die Länge der Eingabe hinausgelangt, sondern stattdessen wieder am anderen Ende beginnt. Da der resultierende überlagerte Zustand wieder ein Quantenzustand ist, muss U_δ^x für alle x, k und q unitär sein.
Wir zerlegen nun δ in zwei Funktionen, eine für die Zustände und eine für die Kopfbewegung:

$$V_b : l_2(Q) \to l_2(Q) \; D : Q \to \{-1, 0, 1\}$$

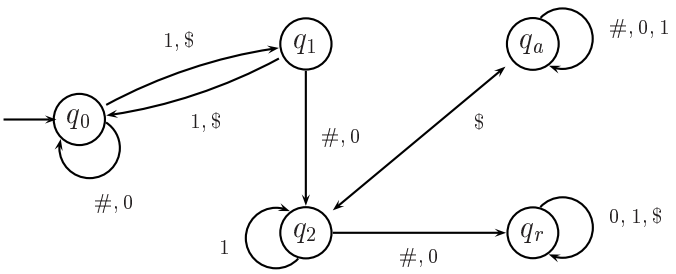
Damit gilt für die Überführungsfunktion

$$\delta(q, b, q', d) = \begin{cases} \langle q'|V_b|q\rangle & \text{falls } D(q') = d \text{ (Amplitude von q')} \\ 0 & \text{falls } D(q') \neq d \end{cases}$$

Um zu überprüfen, ob diese Definition wohlgeformt ist, genügt es zu zeigen, dass V_b unitär ist, was wir an dieser Stelle jedoch nicht tun wollen.
Betrachten wir uns stattdessen lieber ein einfacheres Beispiel. Wir wollen die Sprache 0^*1^* durch einen 2QFA darstellen. Der klassische Automat für diese Sprache wäre sehr einfach:



Allerdings ist dieser Automat nicht als Quantenautomat realisierbar, da er nicht reversibel ist (in Zustand q_2 ist der Vorgängerzustand bei einer gelesenen 0 nicht eindeutig). Stattdessen müssen wir uns eine etwas trickreichere Konstruktion ausdenken, die im folgenden beschrieben wird. Dazu definieren wir zunächst die Mengen $Q = \{q_0, q_1, q_2, q_a, q_r\}$, $Q_a = \{q_a\}$ und $Q_r = \{q_r\}$; die Übergänge sind in der grafischen Repräsentation



In der tabellarischen Definition sind die Übergänge (alle Wahrscheinlichkeiten sind 1):

$V_\#\|q_0\rangle = \|q_0\rangle$	$V_0\|q_0\rangle = \|q_0\rangle$	$V_1\|q_0\rangle = \|q_1\rangle$
$V_\#\|q_1\rangle = \|q_2\rangle$	$V_0\|q_1\rangle = \|q_2\rangle$	$V_1\|q_1\rangle = \|q_0\rangle$
$V_\#\|q_2\rangle = \|q_r\rangle$	$V_0\|q_2\rangle = \|q_r\rangle$	$V_1\|q_2\rangle = \|q_2\rangle$
$V_\#\|q_a\rangle = \|q_a\rangle$	$V_0\|q_a\rangle = \|q_a\rangle$	$V_1\|q_a\rangle = \|q_a\rangle$
$V_\#\|q_r\rangle = \|q_r\rangle$	$V_0\|q_r\rangle = \|q_r\rangle$	$V_1\|q_r\rangle = \|q_r\rangle$
$V_\$\|q_0\rangle = \|q_1\rangle$	$V_\$\|q_1\rangle = \|q_0\rangle$	$V_\$\|q_2\rangle = \|q_a\rangle$
$D(q_0) = 1$	$D(q_1) = -1$	$D(q_2) = 1$
$D(q_a) = 0$	$D(q_r) = 0$	

Für ein Wort $\#0^i1^j\$$ erhalten wir also die Übergangsfolge

$$|1, q_0\rangle \to |2, q_0\rangle \to \ldots \to |i+2, q_0\rangle \xrightarrow{1} |i+1, q_1\rangle \xrightarrow{1} |i+2, q_2\rangle \to \ldots \to |i+j+2, q_2\rangle$$

Von diesem Zustand aus gibt es nun zwei Möglichkeiten: mit $\xrightarrow{\$} |i+j+2, q_a\rangle$ in einen akzeptierenden oder mit $\xrightarrow{0} |i+j+2, q_r\rangle$ in einen verwerfenden Endzustand. Damit erkennt dieser Automat die genannte Sprache.

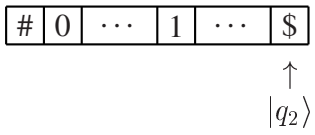
Als etwas komplexeres Beispiel wollen wir nun einen 2QFA für die Sprache $L = \{0^m 1^m \mid m \geq 1\}$ betrachten. Diese Sprache ist nicht regulär. Wir definieren einen Automaten $M_n = \{Q, \Sigma, \delta, q_0, Q_a, Q_r\}$, wobei $Q = \{q_0, q_1, q_2, q_3\} \cup \{r_{j,k} \mid 1 \leq j \leq n,\ 0 \leq k \leq max(j, n-j+1)\} \cup \{s_j \mid 1 \leq j \leq n\}$ (die Anzahl der Zustände hängt also von dem Parameter n ab). Ferner ist $Q_a = \{s_n\}$ und $Q_r = \{q_3\} \cup \{s_j \mid j < n\}$. Die in zwei Funktionen zerlegte Übergangsfunktion von M_n definieren wir folgendermaßen:

$V_\#\|q_0\rangle = \|q_0\rangle$	$V_\$\|q_0\rangle = \|q_3\rangle$
$V_\#\|q_1\rangle = \|q_3\rangle$	$V_\$\|q_2\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \|r_{j,0}\rangle$
$V_\#\|r_{j,0}\rangle = \frac{1}{\sqrt{n}} \sum_{l=1}^{n} e^{\frac{2\pi i}{n} jl} \|s_l\rangle,\ 1 \leq j \leq n$	
$V_0\|q_0\rangle = \|q_0\rangle$	$D(q_0) = 1$
$V_0\|q_1\rangle = \|q_2\rangle$	$D(q_1) = -1$
$V_0\|q_2\rangle = \|q_3\rangle$	$D(q_2) = 1$
$V_0\|r_{j,0}\rangle = \|r_{j,j}\rangle,\ 1 \leq j \leq n$	$D(q_3) = 0$
$V_0\|r_{j,k}\rangle = \|r_{j,k-1}\rangle,\ 1 \leq k \leq j,\ 1 \leq j \leq n$	
$V_1\|q_0\rangle = \|q_1\rangle$	$D(r_{j,0}) = -1,\ 1 \leq j \leq n$
$V_1\|q_2\rangle = \|q_2\rangle$	$D(r_{j,k}) = 0,\ 1 \leq j \leq n,\ k \neq 0$
$V_1\|r_{j,0}\rangle = \|r_{j,n-j+1}\rangle,\ 1 \leq j \leq n$	$D(s_j) = 0,\ 1 \leq j \leq n$
$V_1\|r_{j,k}\rangle = \|r_{j,k-1}\rangle,\ 1 \leq k \leq j \leq n$	

Behauptung 6.2.1 *Sei $w \in \{0,1\}^*$. Dann gilt für alle $n \in \mathbb{N}$*

- $w \in L \Rightarrow M_n$ *akzeptiert die Eingabe mit Wahrscheinlichkeit* 1
- $w \notin L \Rightarrow M_n$ *verwirft die Eingabe mit Wahrscheinlichkeit* $\geq 1 - \frac{1}{n}$

Wie wir gleich sehen werden, wird während der Ausführung des Automaten jedes Wort zweimal gelesen, d.h. die Laufzeit ist linear bezüglich der Länge der Eingabe. Die Anzahl der Zustände von M_n liegt in $O(n^2)$. M_n hält nach $O(n\,|w|)$ Schritten mit Sicherheit.
Der Automat geht in drei Etappen vor, die wir uns nun im einzelnen anschauen wollen.
1. Etappe: In dieser Etappe testet der Automat analog zum vorherigen Beispiel, ob $w \in 0^*\,1^*$, wofür er die Zustände $\{q_0, \ldots, q_3\}$ benötigt. Stellt er hier bereits fest, dass $w \notin 0^*\,1^*$, so verwirft er die Eingabe sofort. Gilt hingegen $w \in 0^+\,1^+$, so endet die Etappe im Zustand $|q_2\rangle$ auf dem Abschlusszeichen \$:



2.Etappe: Hier kommt nun die Regel $V_\$|q_2\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} |r_{j,0}\rangle$ zum Zuge. Bei einer Messung erhalten wir jedes $|r_j, 0\rangle$ mit der Wahrscheinlichkeit $\frac{1}{n}$. Dabei wird auf jeden Fall in der Eingabe ein Schritt nach links vorgenommen und die möglichen Folgezustände sind

$$\begin{aligned} |r_{j,0}\rangle &\xrightarrow{0} |r_{j,j}\rangle \\ |r_{j,0}\rangle &\xrightarrow{1} |r_{j,n-j+1}\rangle \end{aligned}$$

In einem solchen Zustand bleibt die Eingabe stehen und der zweite Index wird dekrementiert, bis er den Wert 0 erreicht. Die Anzahl der Dekrement-Schritte hängt allerdings von der aktuellen Eingabe ab:

- aktueller Eingabewert 0: die Maschine bleibt für j Schritte auf der Zelle stehen und bewegt sich dann einen Schritt nach links
- aktueller Eingabewert 1: die Maschine bleibt für $n - j + 1$ Schritte auf der Zelle stehen und bewegt sich dann einen Schritt nach links

Für ein Eingabewort der Form $0^u\,1^v$ werden also insgesamt (jeweils mit Bewegung nach links) $(j + 1)\,u + (n - j + 2)\,v + 1$ Schritte benötigt. Die Anzahl hängt also von dem zufällig durch die Messung erhaltenen Parameter j ab. Wenn wir uns

jedoch überlegen, was es bedeutet, wenn für zwei verschiedene Parameter j und j' die gleiche Anzahl an Schritten benötigt wird, so gelangen wir zu folgendem Zusammenhang:

$$\begin{aligned} & \text{Für } j \neq j' \text{ gelangt man zur selben Zeit in der Eingabe ganz nach links} \\ \Leftrightarrow \quad & (j+1)\,u + (n-j+2)\,v + 1 = (j'+1)\,u + (n-j'+2)\,v + 1 \\ \Leftrightarrow \quad & j(u-v) = j'(u-v) \\ \Leftrightarrow \quad & u = v \end{aligned}$$

3.Etappe: Falls nun gilt $u = v$, so müssen alle Parameter j gleichzeitig ganz nach links gelangen. Um dies herauszufinden, führen wir eine QFT auf das Ergebnis durch, woraus sich der Übergang $V_{\#}|r_{j,0}\rangle = \frac{1}{\sqrt{n}} \sum_{l=1}^{n} e^{\frac{2\pi i}{n} jl} |s_l\rangle$ ergibt. Damit erreicht man als Endzustand

$$\frac{1}{n} \sum_{j=1}^{n} \sum_{l=1}^{n} e^{\frac{2\pi i}{n} jl} |s_j\rangle.$$

Schließlich führen wir auf diesen Zustand eine Messung durch, d.h. wir messen $|s_j\rangle$ mit Wahrscheinlichkeit $\|\frac{1}{n} \sum_{l=1}^{n} e^{\frac{2\pi i}{n} jl}\|^2$. Damit erhalten wir für $j = n$, dass $|s_n\rangle$ mit Wahrscheinlichkeit 1 gemessen wird; für alle anderen Werte ($j < n$) ergibt sich die Wahrscheinlichkeit aus der geometrischen Summe

$$\frac{1 - e^{\frac{2\pi i}{n} j(n+1)}}{1 - e^{\frac{2\pi i}{n} j}} = 0.$$

Wir erhalten bei der Messung also mit Sicherheit das Ergebnis $|s_n\rangle$, falls $u = v$. Nun stellt sich aber die Frage, mit welcher Wahrscheinlichkeit wir ebenfalls $|s_n\rangle$ messen, falls $u \neq v$. In diesem Fall erreicht genau ein j den linken Rand der Eingabe früher als alle anderen; die Wahrscheinlichkeit, dass es sich nach der QFT um den Zustand $|s_n\rangle$ handelt, ist also

$$Prob(|s_n\rangle) = \|\tfrac{1}{\sqrt{n}} e^{\frac{2\pi i}{n} n}\|^2 = \tfrac{1}{n}.$$

Die Wahrscheinlichkeit, dass der Automat zurecht die Eingabe verwirft, ist somit $1 - \frac{1}{n}$. Damit ist die Behauptung gezeigt.

Anmerkungen: Auf ähnliche Weise lässt sich auch ein Automat für die Sprache $0^m\,1^m\,0^m$ konstruieren, die nicht mal kontextfrei ist. Außerdem wissen wir nun, dass gilt $1QFA \subset REG \subset 2QFA$.

Kapitel 7

Quanten-Kommunikationskomplexität und Kryptologie

7.1 Klassische und probabilistische Varianten

Der Begriff der Kommunikationskomplexität bezeichnet allgemein den Kommunikationsaufwand, der bei einer verteilten Lösung eines Problems entsteht. In der Theorie wird zur Veranschaulichung dabei oft das Bild von zwei oder mehr Spielern benutzt, die eine vorgegebene Aufgabenstellung mit möglichst wenig Kommunikation lösen müssen. Als einfaches Beispiel kann man sich zwei Spieler A und B vorstellen, wobei A einen Bitstring x und B einen Bitstring y erhält mit $x, y \in X \times Y$, $X = Y = \{0,1\}^n$. A soll nun eine berechenbare Funktion $f(x,y)$ berechnen mit $f : X \times Y \to \{0,1\}$. Je nach Gestalt dieser Funktion sind nun verschiedene Fälle für den Kommunikationsaufwand denkbar. Zu beachten ist, dass es bei manchen Funktionen von der Eingabe abhängen kann, was für ein Kommunikationsaufwand entsteht (siehe zweites Beispiel unten); in diesem Fall ist immer die maximale Kommunikationskomplexität gefragt. Beispiele für Funktionen und deren Komplexität sind

- Für $f_1(x,y) = 1 \Leftrightarrow x_1 \oplus x_2 \oplus \ldots \oplus x_n \oplus \overbrace{y_1 \oplus \ldots \oplus y_n}^{s} = 1$ reicht ein Kommunikationsbit aus, da es genügt, wenn B das Bit s an A schickt.
- Für $f_2(x,y) = 1 \Leftrightarrow$ (Anzahl von 1'en in x $=$ Anzahl von 1'en in y) wäre die (worst case) Komplexität $\lceil \log n \rceil$ Bits. In diesem Fall ist es allerdings erforderlich, die Minimalität dieser Anzahl formal zu zeigen, was wir jedoch an dieser Stelle nicht machen werden.

- Für $EQ(x, y) = 1 \Leftrightarrow x = y$ müssen alle n Bits von y übertragen werden, was sich mit einem Widerspruchsbeweis für kleinere Zahlen zeigen lässt.

Dieses letzte Problem lässt sich jedoch mit erheblich weniger Bits lösen, wenn man eine gewisse Fehlerwahrscheinlichkeit zulässt. Dazu geben wir uns eine (möglichst kleine) Konstante ϵ vor und fordern, dass das Problem mit einer Wahrscheinlichkeit von mindestens $1 - \epsilon$ korrekt gelöst wird. Mit Hilfe dieser Zahl ermitteln nun beide Spieler die kleinste Primzahl $p > \frac{n}{\epsilon}$. Aus dem Primzahlsatz folgt dann, dass $p \leq \frac{2n}{\epsilon}$. Daraufhin stellt B mit den Bits von y ein Polynom in der Restklassenmenge $\mathbb{Z}_p$ in folgender Form auf:

$$P(z) = y_1 + y_2\, z + y_3\, z^2 + \ldots + y_n\, z^{n-1}$$

Anschließend wählt B zufällig unter Gleichverteilung eine Zahl $w \in_R \mathbb{Z}_p$, berechnet $P(w) = v$ und schickt v und w an A.
A berechnet ebenfalls ein Polynom $Q(w)$, wobei

$$Q(z) = x_1 + x_2\, z + x_3\, z^2 + \ldots + x_n\, z^{n-1}.$$

Falls nun $Q(w) = v$ ist, so gibt A die Ausgabe 1 aus, falls $Q(w) \neq v$, ist die Ausgabe 0.
Zur Analyse der Fehlerwahrscheinlichkeit genügt es, den Fall $x \neq y$ zu betrachten (wenn $x = y$, so gilt $\forall\, w Q(w) = P(w)$). Formal lässt sich diese Wahrscheinlichkeit schreiben als

$$\text{Prob}_w(Q(w) = P(w) \mid P \neq Q)$$

Da $P \neq Q$, ergibt die Differenz der beiden Polynome erneut ein nichttriviales Polynom vom Grade $\leq n - 1$, das bekanntlich höchstens $n - 1$ verschiedene Nullstellen besitzt. Damit kann man nun die Wahrscheinlichkeit abschätzen:

$$\text{Prob}_w(Q(w) - P(w) = 0) \leq \tfrac{n-1}{p} < \tfrac{n}{\frac{n}{\epsilon}} = \epsilon$$

Somit liefert diese Methode mit der geforderten Wahrscheinlichkeit das korrekte Ergebnis. Die Komplexität ist in diesem Fall gegeben durch die Länge von w und v:

$$|w| \leq \log p + 1 \leq \log(\tfrac{2n}{\epsilon}) + 1 = 2 + \log n + \log \tfrac{1}{\epsilon}$$

Die Komplexität liegt hier also bei $O(\log n)$. Wenn A und B vor dem Spiel eine gemeinsame Zufallsfolge von Bits austauschen, aus der sich dann für beide dieselbe Zufallszahl w reproduzieren ließe, könnte man die Kommunikation auf den Transfer von v reduzieren. Wir werden zum Schluss dieses Kapitels noch eine weitere Variante für dieses Problem kennenlernen.

Definition 7.1.1 *Sei P ein Kommunikationsproblem und $\epsilon \geq 0$ eine Konstante. Dann bezeichnet der Ausdruck $C_\epsilon(P)$ das Minimum m, so dass es ein probabilistisches Protokoll mit weniger als oder genau m Kommunikationsbits für P gibt, das mit einer Wahrscheinlichkeit größer als $1 - \epsilon$ korrekt ist.*

Als Sonderfall bezeichnet $C_0(P)$ die deterministische Lösung. Da das Minimum in den meisten anderen Fällen von ϵ abhängt, notieren wir $C_\epsilon(P)$ mit Hilfe der O-Notation, so dass der genaue Wert von ϵ gar keine Rolle mehr spielt, und bezeichnen dies mit $C(P)$. Für das obige Beispiel gilt dann $C_0(EQ) = n$, $C(EQ) = O(\log n)$.

Für ein weiteres Problem, die Funktion

$$Disj(x, y) = \bigvee_{i \in 1 \ldots n} (x_i \wedge y_i)$$

lässt sich zeigen, dass $C(Disj) = \Omega(n)$.
Ein Problem, das wir später noch näher betrachten werden, ist

$$EQ'(x, y) = \begin{cases} 1 & \text{falls } \Delta(x, y) = 0 \\ 0 & \text{falls } \Delta(x, y) = \frac{n}{2} \\ undef & \text{sonst} \end{cases}$$

Dabei ist $\Delta(x, y) = \|\{i \,|x_i \neq y_i\}\|$ der Hamming-Abstand zwischen x und y. Für dieses Problem gilt ebenfalls $C_0(EQ') = \Omega(n)$.

7.2 Quanten-Kommunikationskomplexität

Wir wollen nun einige Verfahren betrachten, mit denen durch die Anwendung von Qubits Vorteile in der Kommunikation erzielt werden können. Dabei gibt es zwei verschiedene Modelle, auf welche Weise man Qubits ins Spiel bringen kann:

1. Spieler A und B schicken sich nur klassische Bits und können lediglich vor dem Spiel Quantenzustände teilen, die verschränkt sind.

2. Die Spieler dürfen Qubits übertragen.

Auf den ersten Blick erstaunlich erscheint, dass das erste Modell mächtiger ist als das zweite. Allerdings weiß man aus der Quantenteleportation, dass ein Qubit mit zwei klassischen Bits und einem gemeinsamen Zustand übertragen werden kann. Außerdem gibt es die *Holevo-Schranke*, die besagt, dass es nicht möglich ist, mit n

Qubits mehr als n klassische Informationsbits zu übertragen.

Betrachten wir gleich ein konkretes Beispiel: 3 Spieler A, B und C besitzen jeweils ein x_A, x_B und x_C mit $x_i \in \{0,1\}$ und $x_A \oplus x_B \oplus x_C = 0$. Zugelassen sind also die Kombinationen $(0,0,0)$, $(0,1,1)$, $(1,0,1)$ und $(1,1,0)$, die jeweils mit Wahrscheinlichkeit $\frac{1}{4}$ auftreten sollen (wir nehmen eine Gleichverteilung an). Die Aufgabe ist nun, dass jeder Spieler ein Bit a, b bzw. c wählt, so dass gilt $a \oplus b \oplus c = x_A \vee x_B \vee x_C$. Wir betrachten dabei zwei Fälle: in a) können A, B und C eine gemeinsame Zufallsfolge lesen, in b) dürfen A, B und C vor dem Spiel Quantenzustände teilen.
Fall a): Wir bezeichnen das Bit von A mit a_0, falls $x_A = 0$ und mit a_1, falls $x_A = 1$; analog definieren wir b_0, b_1, c_0 und c_1 für die Spieler B bzw. C. Durch eine Gegenüberstellung aller vier Möglichkeiten für die x_A, x_B und x_C und der gewünschten Resultate können wir feststellen, dass das Protokoll höchstens mit einer Wahrscheinlichkeit von $\frac{3}{4}$ korrekt sein kann:

$$\begin{array}{rcl} a_0 \oplus b_0 \oplus c_0 & \stackrel{!}{=} & 0 \\ a_0 \oplus b_1 \oplus c_1 & \stackrel{!}{=} & 1 \\ a_1 \oplus b_0 \oplus c_1 & \stackrel{!}{=} & 1 \\ a_1 \oplus b_1 \oplus c_0 & \stackrel{!}{=} & 1 \\ \hline \oplus \quad 0 \oplus 0 \oplus 0 & \stackrel{!}{=} & 1 \end{array}$$

Da die spaltenweise Addition mittels XOR einen Widerspruch ergibt, wird klar, dass mindestens ein Wert falsch sein muss und die besagte obere Schranke für die Wahrscheinlichkeit gilt. Auch bei einer Erweiterung der Methode durch eine zufällige Auswahl r der Elemente erhält man keine Verbesserung, da hier mit der totalen Wahrscheinlichkeit gilt (D bezeichne das Protokoll):

$$\begin{aligned} Prob_{r,x_A,x_B,x_C}(D \text{ korrekt}) &= \sum_r Prob_{x_A,x_B,x_C}(D_r \text{ korrekt})\, Prob(\text{Zufallsfolge } r) \\ &\leq \frac{3}{4} \sum_r Prob(\text{Zufallsfolge } r) \\ &\leq \frac{3}{4} \end{aligned}$$

Fall b): Die Spieler teilen sich hier den Quantenzustand

$$\psi = \tfrac{1}{2}\,|000\rangle - |011\rangle - |101\rangle - |110\rangle),$$

der unter dem Namen Greenberger-Horne-Zeilinger-Zustand (GHZ) bekannt ist. Die drei Spieler verwenden alle die gleiche Strategie, welche hier exemplarisch an Spieler A dargestellt werden soll: Falls $x_A = 0$, wird Qubit 1 bezüglich der

Standard-Basis ($|0\rangle$, $|1\rangle$) gemessen und das Ergebnis der Messung (0 oder 1) direkt ausgegeben. Gilt $x_A = 1$, so wird Qubit 1 bezüglich der Hadamard-Basis gemessen ($|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$); dabei wird eine 0 ausgegeben, wenn man $+$ misst und eine 1, wenn man $-$ misst. Analog dazu verfahren auch die Spieler B und C bei der Messung von Qubit 2 und 3. Wir betrachten nun alle möglichen Fälle, die wir oben für die x_i festgestellt haben:

- Bei der Kombination $(0,0,0)$ messen alle Spieler den GHZ-Zustand bezüglich der Standard-Basis und erhalten so als mögliche Ergebnisse der Messungen 000, 011, 101 und 110 für abc. Da die Parität aller vier Ergebnisse gerade ist, gilt hier immer $a \oplus b \oplus c = 0 = x_A \vee x_B \vee x_C$.

- Für die anderen Fälle $(0,1,1)$, $(1,0,1)$ und $(1,1,0)$ für die x_i werden jeweils zwei der Qubits bezüglich der Hadamard-Basis gemessen und der GHZ-Zustand geht über in

$$\begin{aligned} & \frac{1}{2}\left(\frac{1}{2}(|+\rangle + |-\rangle)(|+\rangle + |-\rangle)|0\rangle - \frac{1}{2}(|+\rangle + |-\rangle)(|+\rangle - |-\rangle)|1\rangle\right. \\ - \quad & \left.\frac{1}{2}(|+\rangle - |-\rangle)(|+\rangle + |-\rangle)|1\rangle - \frac{1}{2}(|+\rangle - |-\rangle)(|+\rangle - |-\rangle)|0\rangle\right) \\ = \quad & \frac{1}{2}\Big(\underbrace{|+-0\rangle}_{\text{Ausgabe } 010} + \underbrace{|-+0\rangle}_{\text{Ausgabe } 100} - \underbrace{|++1\rangle}_{\text{Ausgabe } 001} + \underbrace{|--1\rangle}_{\text{Ausgabe } 111} \Big) \end{aligned}$$

  Die zugehörige Ausgabe hierzu ist also stets von ungerader Parität und liefert somit immer $a \oplus b \oplus c = 1 = x_A \vee x_B \vee x_C$.

Wir wollen nun eine Anwendung dieser Methode betrachten, die den Vorteil für die Quantenkommunikation zeigt. Seien x, y, $z \in \{0,1,2,3\}$ mit Binärdarstellung $x = x_1x_0$, $y = y_1y_0$ und $z = z_1z_0$, die die Bedingung $x + y + z \, mod \, 2 = 0$ erfüllen sollen. Ferner definieren wir die Funkion

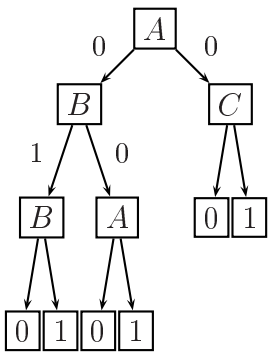
$$f(x,y,z) = \tfrac{x+y+z \, mod \, 4}{2}.$$

Wir wollen nun den Kommunikationsaufwand analysieren, der für die Berechnung von f erforderlich ist. Wir betrachten ein 3-Spieler Kommunikationsmodell, in dem erforderlich ist, dass nach der Kommunikation alle drei Spieler die Funktion berechnen können. Wir betrachten zum einen den Fall a), dass die Spieler Quantenzustände teilen, und danach den Fall b), in dem keine Quantenzustände verwendet werden.
Fall a): Wenn man sich die binäre Addition vor Augen hält, wird klar, dass $x + y + z \, mod \, 2 = 0 \Leftrightarrow x_0 \oplus y_0 \oplus z_0 = 0$. Ebenso erhält man damit eine einfachere Darstellung der Funktion $f(x,y,z) = x_1 \oplus y_1 \oplus z_1 \oplus (x_0 \vee y_0 \vee z_0)$, wobei die Klammer den Übertrag bei der binären Addition darstellt. Mit dem zuvor beschriebenen Protokoll auf Basis von Quantenzuständen können die Spieler

A, B und C die Bits a, b und c ermitteln, wobei sie in diesem Fall die Werte von x_0 als x_A, y_0 als x_B und z_0 als x_C verwenden. Dann gilt wie oben gesehen $a \oplus b \oplus c = x_A \vee x_B \vee x_C = x_0 \vee y_0 \vee z_0$. Somit braucht Spieler A nur das Bit $x_1 \oplus a$, Spieler B das Bit $y_1 \oplus b$ und Spieler C das Bit $z_1 \oplus c$ zu schicken. Damit können alle drei Spieler die Funktion rekonstruieren:

$$\begin{aligned}(x_1 \oplus a) \oplus (y_1 \oplus b) \oplus (z_1 \oplus c) &= x_1 \oplus y_1 \oplus z_1 \oplus (a \oplus b \oplus c) \\ &= x_1 \oplus y_1 \oplus z_1 \oplus (x_0 \vee y_0 \vee z_0) \\ &= f(x, y, z)\end{aligned}$$

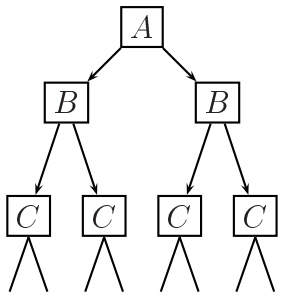
Somit genügen in diesem Fall drei Kommunikationsbits für die Übertragung der für die Funktion relevanten Information.
Fall b): Für die Überlegungen zu diesem Punkt greifen wir auf die Darstellung eines Protokolls in Form eines Baums zurück. Innere Knoten des Baums werden mit der Spielerbezeichnung beschriftet, in diesem Falle also A, B oder C; Blattknoten erhalten den Wert 0 oder 1, der den Funktionswert $f(x, y, z)$ repräsentiert. Die Verbindungslinien zwischen den Knoten signalisieren, welcher Bitwert zwischen den Knoten übermittelt wird. Aus einem solchen Baum lässt sich nun die Anzahl der Bits über die Maximaltiefe des Baums ermitteln. Ein anschauliches Beispiel für einen Protokollbaum wäre folgender:



Wir stellen nun noch ein Lemma für unser konkretes Beispiel auf:

Lemma 7.2.1 *Sei P ein Protokoll für f mit zugehörigem Baum B_P. Dann gilt: in jedem Pfad von der Wurzel zu einem Blattknoten müssen Knoten für A, B und C vorkommen.*

Wenn wir ohne Beschränkung der Allgemeinheit Spieler A als Wurzel des Baums festlegen, so könnte der Baum für unser konkretes Beispiel folgende Form haben:



Da jeder Spieler laut Problemstellung zwei Bits besitzt, kommen als mögliche Eingaben für A die Werte 00, 01, 10 und 11 in Frage. Wir trennen die Menge der möglichen Eingaben von A nun in s_0 als die Eingaben, die das Kommunikationsbit 0 produzieren, und s_1 als die Eingaben, die das Kommunikationsbit 1 produzieren. Für die weiteren Überlegungen nehmen wir wiederum o.B.d.A. an, dass $00 \in s_0$. Außerdem betrachten wir zwei Fälle:
Fall 1 ($|s_0| = 1$): Falls $x \in s_1$, $y = 01$ und $z = 00$, so sind die möglichen Kombinationen $(01, 01, 00)$ und $(11, 01, 00)$ (die dritte Möglichkeit entfällt, da die Summe durch 2 teilbar sein muss). Für das erste dieser Tripel liefert die Funktion f eine 1, für das zweite eine 0 zurück; somit ist durch das gesendete Kommunikationsbit weder für B noch für C ersichtlich, welcher Funktionswert sich ergibt, und das Protokoll kann nicht korrekt sein.
Fall 2 ($|s_0| \geq 2$): Es gelte z.B. $s_0 = \{00, 01\}$. A besitze ein $x \in s_0$ und C den Wert $z = 00$; der Wert von y sei unbekannt. Damit sind die Kombinationen $(00, 00, 00)$, $(00, 10, 00)$, $(01, 01, 00)$ und $(01, 11, 00)$ möglich, für die die Funktion f die Werte 0, 1, 1 und 0 liefert. Damit ist f für die Werte 10 und 01 von y erfüllt und für 00 sowie 11 nicht erfüllt. Wenn wir nun jedoch die Kombinationen für $z = 01$ betrachten, erhalten wir als mögliche Kombinationen $(01, 00, 01)$, $(01, 10, 01)$, $(00, 01, 01)$ und $(00, 11, 01)$; die Werte von f hierfür sind 1, 0, 1 und 0. Hier ist f damit für die Werte 00 und 01 erfüllt und für 11 und 10 nicht erfüllt, d.h. für andere Wertepaare als für $z = 00$. Somit muss auch dieses Protokoll falsch sein. Durch die Analyse von den weiteren Möglichkeiten für die Struktur vom Kommunikationsbaum kann man zeigen, dass das Problem grundsätzlich nicht auf Basis von drei klassischen Kommunikationsbits lösbar ist, es sind also mindestens 4 Bits notwendig. Mit der Quantenlösung erreicht man also eine sichtbare Verbesserung.

Betrachten wir nun ein allgemeines Modell einer Funktion aus der Menge

$$\mathcal{F}_n : \left\{ f : \{0,1\}^n \to \{0,1\} = \{0,1\}^N \right\}, N = 2^n$$

Hinter der Darstellung durch $\{0,1\}^N$ steckt die Idee, dass man jede Boolesche Funktion mit n Eingabebits als Array von Bits der Länge 2^n darstellen kann, indem

man die Funktionswerte nach der lexikographisch sortierten Eingabe hintereinander schreibt (also von $f(0^n), f(0^{n-1}1)$ bis $f(1^n)$).

Definition 7.2.2 *Sei $L : \{0,1\}^2 \to \{0,1\}$, $g, h \in \mathcal{F}_n$ und sei $x \in \{0,1\}^n$. Wir definieren die Funktion $L(g,h) : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ als*

$$L(g,h)(x) = L(g(x), h(x)).$$

Satz 7.2.3 *Sei $F : \mathcal{F} \to \{0,1\}$ und $L : \{0,1\} \times \{0,1\} \to \{0,1\}$. Falls es einen Quanten-Algorithmus A der Form $|x\rangle|y\rangle \to |x\rangle|f(x) \oplus y\rangle$ für $F(f)$ gibt, der t Fragen an f stellt, 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.*
Falls der Algorithmus A mit Wahrscheinlichkeit α korrekt ist, so ist dies das Protokoll auch.

Wir wollen diesen Satz gleich im Anschluss beweisen, uns zunächst aber einigen einfachen Beispielen für die Problematik zuwenden.
Beispiel 1: $F = OR$, $L = AND$. Für die Oder-Funktion kann der Algorithmus von Grover verwendet werden, der $O(\sqrt{N})$ Fragen an f stellt. Damit können für g und h die Eingaben x ermittelt werden, für die die Funktion den Wert 1 liefert. Zusammen mit der Und-Funktion ergibt sich dann, dass $F(L(g,h)) = 1 \Leftrightarrow$ (es gibt ein x mit $g(x) = h(x)$) und somit $F(L(g,h)) = Disj(g,h)$, für die es ein Protokoll mit $t = O(\log N)$ gibt. Insgesamt gibt es also ein Quantenkommunikationsprotokoll mit $O(\sqrt{N}\log N)$ Quantenkommunikationsbits für dieses Problem.
Beispiel 2: $F = OR'$, d.h.

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

was mit dem Quantenalgorithmus von Deutsch-Josza mit $t = 1$ Fragen gelöst werden kann. Für L gelte ferner $L = XOR$, was bedeutet, dass

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

und damit $F(L(g,h)) = EQ'(g,h)$. Für EQ' gibt es ein Protokoll mit Komplexität $O(\log N)$, was somit gleichzeitig auch die Gesamtkomplexität für das Beispielproblem ist. Man kann zeigen, dass klassisch für dieses Problem eine untere Schranke von $\Omega(N)$ Kommunikationsbits gilt. Das zeigt einen exponentiellen Unterschied

zwischen den Quanten und klassischen Kommunikationskomplexitäten.

Beweis zum Satz: Sei G ein Quantenschaltkreis für $F(f)$ mit $f = L(g,h)(x) = L(g(x),h(x))$; der Algorithmus A simuliere diesen Schaltkreis. Fragen an f lassen sich somit folgendermaßen darstellen

$$|x\rangle\,|y\rangle \rightarrow |x\rangle\,|f(x)\oplus y\rangle \text{ mit } f(x) = L(g(x),h(x))$$

Damit lässt sich nun ein allgemeines Verfahren für die Kommunikation beschreiben:

1. Spieler A: $|x\rangle\,|y\rangle\,|0\rangle \rightarrow \overbrace{|x\rangle\,|y\rangle\,|g(x)\rangle}^{\text{Kommunikation an } B}$, was insgesamt $n+2$ Quantenkommunikationsbits ergibt (x hat die Dimension n, y und $g(x)$ sind jeweils 1 Qubit).

2. Spieler B: $|x\rangle\,|y\rangle\,|g(x)\rangle \rightarrow \overbrace{|x\rangle\,|L(g(x),h(x))\oplus y\rangle\,|g(x)\rangle}^{\text{Kommunikation an } A}$, was ebenfalls $n+2$ Quantenkommunikationsbits sind.

3. Schließlich führt Spieler A die Transformation

$$|x\rangle\,|L(g(x),h(x))\oplus y\rangle\,|g(x)\rangle \rightarrow |x\rangle\,|L(g(x),h(x))\oplus y\rangle\,|0\rangle = |x\rangle\,|f(x)\oplus y\rangle\,|0\rangle$$

   durch und erhält somit den Wert von $L(g,h)(x)$.

Wenn für den Quantenalgorithmus also t Fragen an f notwendig sind, so ergibt sich mit diesem Verfahren insgesamt eine Kommunikationskomplexität von $t(2n+4)$ Bits. □

7.3 Finger Printing

Der Begriff Finger Printing beschreibt allgemein die Idee, Elemente aus einer sehr umfangreichen Struktur durch eine Art Fingerabdruck auf sehr viel kleinere und somit effizienter handhabbare Strukturen abzubilden. Ein häufiges Verfahren ist Hashing, was z.B. mit der Operation $mod\ q$ für eine geeignet gewählte Zahl q funktioniert.
Eine solche Methode würde sich für das bereits angesprochene Problem $EQ(x,y)$ anbieten: Spieler A besitze ein $x \in \{0,1\}^n$ und Spieler B ein $y \in \{0,1\}^n$; beide sollen nun einem dritten Spieler C die notwendige Information übermitteln, dass dieser bestimmen kann, ob $x = y$ gilt. Im deterministischen Fall, der mit Sicherheit

das richtige Ergebnis liefert, müssen jeweils alle n Stellen von x bzw. y verglichen und damit auch als Kommunikationsbits übertragen werden. Auch mit Komprimierungsverfahren erreicht man im worst case keine Verbesserung. Geht man probabilistisch vor, so lässt man eine Fehlerwahrscheinlichkeit von $\epsilon > 0$ zu und erlaubt A und B, eine gemeinsame Folge von Zufallsbits zu verwenden; damit ließe sich die Kommunikationskomplexität auf $O(\log n)$ senken. Eine bessere Lösung erreicht man jedoch mit Codes der Form $E : \{0,1\}^n \to \{0,1\}^m$, die das Eingabewort x in $E(x)$ transformieren und damit z.B. die Zeichenfolge 010 in die Zeichenfolge 000111000 übertragen (was einer fehlertoleranten Übertragung dienen würde).
Für das EQ-Problem geeignet ist beispielsweise der *Justesen-Code*, von dem wir nur die Charakteristiken darstellen wollen, ohne auf Einzelheiten der Implementierung einzugehen. Bei diesem Code ist die Zieldimension $m = c\,n$ mit $c > 2$. Das besondere an diesem Code ist nun, dass stets die Implikation $x \neq y \Rightarrow \Delta(E(x), E(y)) \geq (1-\delta)\,m$ gilt für $0 < \delta < \frac{9}{10} + \frac{1}{15c}$.
Damit können wir ein mögliches Protokoll für $EQ(x,y)$ definieren:

- Eingabe x, $y \in \{0,1\}^n$, Anwendung von $E : \{0,1\}^n \to \{0,1\}^m$
- A und B wählen aus der gemeinsamen Zufallsfolge ein $i \in_R \{1,\ldots,m\}$
- A schickt $E_i(x)$, B schickt $E_i(y)$ an C
- Antwort von C ist ja, falls $E_i(x) = E_i(y)$, andernfalls ist die Antwort nein

Dieses Protokoll liefert auf jeden Fall die richtige Lösung, falls $x = y$, denn dann gilt $E_i(x) = E_i(y)$ und C antwortet richtig. Falls jedoch $x \neq y$, gilt wegen den Eigenschaften des Justesen-Codes, dass

$$\sum_i Prob(E_i(x) \neq E_i(y)) \geq \tfrac{(1-\delta)m}{m} = 1-\delta,$$

womit die Fehlerwahrscheinlichkeit durch $Prob(Fehler) \leq \delta$ abgeschätzt werden kann. Soll diese Fehlerwahrscheinlichkeit nun $< \epsilon$ für eine vorgegebene Konstante sein, so muss man insgesamt k Stellen von $E(x)$ und $E(y)$ vergleichen, damit $\delta^k < \epsilon$ ist. Damit erhält man eine Kommunikationskomplexität, die in $O(\log(\frac{1}{\epsilon}))$ liegt.
Das Problem bei dieser Vorgehensweise ist die gemeinsame Folge von Zufallszahlen, die letztendlich auch auf einem Austausch von Information beruht. Mit verschiedenen Zufallsfolgen für A und B würde man eine Komplexität von $\theta(\sqrt{n})$ erhalten, was jedoch sehr schwierig zu zeigen ist.
Eine Verbesserung kann jedoch auch hier wieder mit einer Quantenlösung erreicht werden, deren Idee es ist, die Eingaben x und y in Quantenzustände $|f(x)\rangle$ und $|f(y)\rangle$ zu transformieren und damit die Beziehung $x \neq y \Rightarrow \langle f(x)|f(y)\rangle \approx 0$ zu erhalten. Das Approximationszeichen bedeutet dabei, dass die Quantenzustände

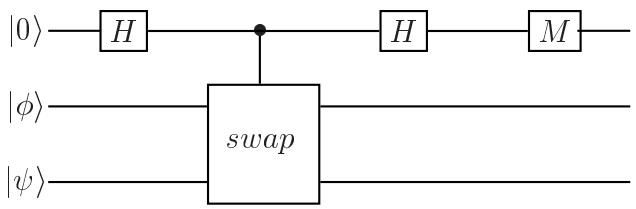
„fast senkrecht“ aufeinander stehen, also das innere Produkt deutlich von 1 verschieden ist. Als Funktion f verwenden wir den Justesen-Code, so dass wir eine Transformation der Gestalt

$$x \to |h_x\rangle = \tfrac{1}{\sqrt{m}} \sum_{i=1}^{m} |i\rangle\,|E(i)\rangle$$

erhalten. Analog gehen wir auch für y vor. Damit gilt dann

- $x = y \Rightarrow \langle h_x|h_y\rangle = 1$
- $x \neq y \Rightarrow \langle h_x|h_y\rangle \leq \frac{1}{m} \underbrace{\delta m}_{\text{Anzahl gleicher Bits}} = \delta$

A und B schicken nun $|h_x\rangle$ und $|h_y\rangle$ an C. C benutzt zur Ermittlung des Ergebnisses folgenden Quantenschaltkreis:



Dabei gilt $|\phi\rangle = |h_x\rangle$ und $|\psi\rangle = |h_y\rangle$, was beides Quantenzustände mit jeweils $m + 1 = O(\log n)$ Qubits sind. Die swap-Operation bewirkt eine Vertauschung $swap(|\phi\rangle|\psi\rangle) = |\psi\rangle|\phi\rangle$. Schrittweise betrachtet ergeben sich folgende Übergänge:

$$\begin{aligned}
|0\rangle|\phi\rangle|\psi\rangle &\overset{H}{\to} \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)|\phi\rangle|\psi\rangle \\
&\overset{swap}{\to} \frac{1}{\sqrt{2}}\left(|0\rangle|\phi\rangle|\psi\rangle + |1\rangle|\psi\rangle|\phi\rangle\right) \\
&\overset{H}{\to} \frac{1}{2}\left(|0\rangle(|\phi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle) + |1\rangle(|\phi\rangle|\psi\rangle - |\psi\rangle|\phi\rangle)\right)
\end{aligned}$$

Wenn wir nun die Quantenzustände darstellen als $|\phi\rangle = \sum_i a_i\,|i\rangle$ und $|\psi\rangle = \sum_i b_i\,|i\rangle$ mit $a_i, b_i \in (0, \frac{1}{\sqrt{m}})$ und $\sum a_i^2 = \sum b_i^2 = 1$, so lässt sich im letzten Ausdruck die Amplitude des Zustands $|1\rangle$ auch ausdrücken als

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

Die Wahrscheinlichkeit ergibt sich dann aus dem Amplitudenquadrat

$$\begin{aligned}\frac{1}{4}\sum_{i,j}|a_i\,b_j - a_j\,b_i|^2 &= \frac{1}{4}\sum_{i,j}((a_i\,b_j)^2 + (a_j\,b_i)^2 - 2a_i\,b_j\,a_j\,b_i)\\ &= \frac{1}{2}\,(1 - \sum_i a_i\,b_i\ \sum_j a_j\,b_j)\\ &= \frac{1}{2}\,(1 - \langle\phi|\psi\rangle^2)\end{aligned}$$

Ist nun $x = y$, so sind die Quantenzustände identisch und ihr inneres Produkt ergibt 1, so dass der Ausdruck 0 ergibt. Ist hingegen $x \neq y$, so gilt

$$\tfrac{1}{2}\,(1 - \langle\phi|\psi\rangle^2) \geq \tfrac{1}{2}\,(1 - \delta^2) = \epsilon'$$

Somit macht C seine Antwort von der Messung abhängig: ergibt die Messung den Wert 1, so liefert C das Ergebnis $x \neq y$, ergibt die Messung 0, so geht C von $x = y$ aus. Diese Vorgehensweise beinhaltet eine Fehlerwahrscheinlichkeit von $1 - \epsilon'$, so dass wir $O(\log(\frac{1}{\epsilon}))$ Wiederholungen machen müssen, um den Fehler kleiner als ϵ zu bekommen.

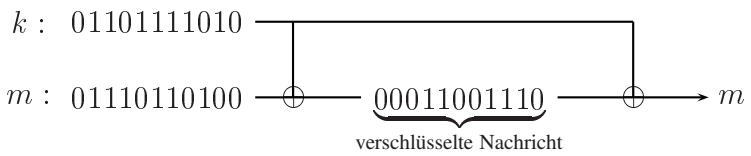
Ingesamt haben wir für dieses Problem also vier verschiedene Kommunikationsprotokolle kennengelernt, die wir hier noch einmal zusammenfassend darstellen wollen:

- Deterministisch ergibt sich stets eine Kommunikationskomplexität von $O(n)$
- Bei probabilistischer Vorgehensweise mit gemeinsamer Zufallsfolge wird immer nur ein Bit verschickt, was eine Kommunikationskomplexität von $O(1)$ ergibt
- Ohne gemeinsame Zufallsfolge liefert die probabilistische Variante eine Kommunikationskomplexität von $\theta(\sqrt{n})$
- Bei der Quantenlösung erreicht man eine Quantenkommunikationskomplexität von $O(\log n)$

7.4 Quantenkryptologie

In der Kryptologie geht es allgemein darum, Information geschützt von einem Kommunikationspartner A (meist Alice) und einem anderen Kommunikationspartner B (Bob) zu übertragen. Dazu wird zumeist die symmetrische Verschlüsselung verwendet, bei der der Sender A eine Nachricht m und einen Schlüssel k mit einer Verschlüsselungsfunktion $f(m, k)$ in eine verschlüsselte Nachricht transformiert, die

dann an B übertragen wird. B verwendet zur Entschlüsselung eine inverse Funktion f^*, mit der die Nachricht wiederhergestellt werden kann: $f^*(f(m,k),k) = m$. In diesem Verfahren kommt es vor allem auf die Verschlüsselungs- bzw. Entschlüsselungsfunktion an, zu einem gewissen Anteil aber auch auf den Schlüssel. Die maximale Sicherheit erreicht man mit sogenannten *One-Time-Pads*, die einmalig verwendete Schlüssel derselben Länge wie die Nachricht darstellen. Als Funktion wird dabei in beiden Richtungen einfach die bit-weise XOR-Verknüpfung verwendet:



Formal gilt also $f(m,k) = m \oplus k = m'$ und $f^*(m',k) = m' \oplus k = m$. Dieses Verfahren ist deshalb sehr sicher, weil jedes Bit quasi einzeln kodiert wird und somit die Wahrscheinlichkeit für einen bestimmten Bit-Wert jeweils $\frac{1}{2}$ beträgt, wenn man den Schlüssel wirklich nur einmal verwendet.
Ein schwieriger Punkt bei der One-Time-Pad Methode ist allerdings die Übergabe oder Vereinbarung des Schlüssels; wurde die Übertragung des Schlüssels bereits abgehört, nützt das Verfahren nichts mehr. In diesem Zusammenhang gibt es ein Schlüsselvereinbarungsprotokoll für einen Schlüssel der Länge n auf Basis von Qubits, das wir uns nun näher anschauen wollen.
Sender A führt folgende Schritte durch:

Wähle $a, b \in_R \{0,1\}^{4n}$
FOR i=1 TO 4n DO

$$|q_i\rangle = \begin{cases} |a_i\rangle & \text{falls } b_i = 0 \\ H\,|a_i\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{a_i}|1\rangle\right) & \text{falls } b_i = 1 \end{cases}$$

Schicke $|q_1\rangle \ldots |q_{4n}\rangle$ an B

Empfänger B geht nun so vor:

Wähle $c \in_R \{0,1\}^{4n}$
FOR i=1 TO 4n DO
falls $c_i = 0$: Messung von q_i bezüglich Standard-Basis
falls $c_i = 1$: Messung von q_i bezüglich Hadamard-Basis

Somit entscheiden die zufällig gewählten b_i und c_i über die Kodierung und Dekodierung, indem sie die Basis für die Messung festlegen. Um die Kodierung und Dekodierung abzugleichen, schickt B nun c an A; umgekehrt schickt A seinen Vektor b an B.
Sei nun $G = \{i \,|\, c_i = b_i\}$ die Menge der Komponenten, für die die Vektoren b und c das gleiche Bit haben. Der Erwartungswert der Größe dieser Menge ist $E[|G|] = 2n$. A wählt eine Untermenge $G' \subset_R G$ mit $|G'| = n$ und schickt alle Paare $\{(a_i, i) \,|\, i \in G'\}$ an B; genauso schickt B alle Paare $\{(q_i, i) \,|\, i \in G'\}$ an A. Wenn nun für alle $i \in G'$ gilt, dass $a_i = q_i$, so gilt die Verbindung als sicher und es wird als Schlüssel die Folge der $\{q_j \,|\, j \in G \backslash G'\}$ gewählt. Wenn es hingegen ein i geben sollte, für das $a_i \neq q_i$, so muss davon ausgegangen werden, dass die Verbindung abgehört wurde und der Schlüssel wird verworfen.
Die Erklärung dafür ist, dass das „Abhören“ nur durch eine Messung an den $|q_i\rangle$ erfolgen kann; ein uneingeweihter Dritter C müsste die Basis der Messung bei jedem Qubit raten und würde jeweils nur mit Wahrscheinlichkeit $\frac{1}{2}$ die richtige Wahl treffen. Wenn z.B. die Basis bei der Messung von $|q_i\rangle$ für ein $i \in G'$ falsch gewählt wird, so bemerken A und B dies sofort, da dann die obige Bedingung $a_i \neq q_i$ erfüllt ist. Die Wahrscheinlichkeit, dass C mithört, ohne dass A und B es bemerken, entspricht demnach der Wahrscheinlichkeit, dass C auf den Positionen in G' stets die richtige Basis wählt und ist somit $P = \frac{1}{2}^n$.

Literaturverzeichnis

Dorit Aharonov, Lecture Notes.
http://www.cs.huji.ac.il/ doria/qcomp.huji2001.html

Jozef Gruska, *Quantum Computing*, McGraw-Hill, 1999.

Michael Nielsen and Isaac Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

John Preskill, Lecture Notes.
http://www.theory.caltech.edu/people/preskill/ph229/

Umesh Vazirani, Lecture Notes
http://www.cs.berkeley.edu/ vazirani/quantum.html

Los Alamos Archive
http://xxx.lanl.gov/archive/quant-ph