Seminar-Vortrag im Bereich Quanten-Informationstheorie: **Quantenalgorithmen**

Inhaltsverzeichnis

1 Einführung **2**
- 1.1 Dualsystem . . . 2
- 1.2 Operatoren . . . 3
- 1.3 Circuit- Diagramme . . . 6

2 Die Born Regel **8**

3 Anwendungen **9**
- 3.1 allgemeines . . . 9
- 3.2 Deutsch's Problem . . . 10
- 3.3 Bernstein-Problem . . . 11
- 3.4 Simon's Problem . . . 13

4 Zusammenfassung **14**

5 Quellenangabe **14**

Quantenalgorithmen

von Manuel Matt

1 Einführung

Mit sogenannten Quantenalgorithmen versucht man wie mit „klassischen“ Algorithmen mit gegebenen Operatoren und möglichst wenig Rechenaufwand eine Problemstellung mathematisch zu lösen. Der Unterschied zu klassischen Algorithmen liegt in der durch quantenmechanische Eigenschaften bedingten Erweiterung der Menge der uns zur Verfügung stehenden Operatoren und deren Eigenschaften.
Im folgenden werde ich die Wichtigsten von ihnen Vorstellen. Weiterhin werde ich Circuit - Diagramme einführen, die der Veranschaulichung von Operator-Relationen und im letzten Teil einiger Anwendungen dienen sollen.
Nach der Einführung werde ich kurz auf die in der Quantenmechanik wichtige Born- und verallgemeinerte Born- Regel eingehen. Diese werden auch eine sehr große Rolle für unsere Quantenalgorithmen spielen.

1.1 Dualsystem

Alle Computer, sowohl die klassischen, wie auch die Quanten-Computer arbeiten im Dualsystem (Binärsystem). Deshalb ist es wichtig sich kurz mit der üblichen Schreibweise vertraut zu machen:
Es wird jede Zahl durch eine Folge von 0 und 1 dargestellt, so ergibt sich Beispielsweise für die Zahl 5 (Dezimal) die Zahl 101 (Dual). Ein Quantenmechanischer Zustand (auch möglich für einen nicht QM- Zustand), wird durch die Vektoren

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ und } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

. dargestellt Weiterhin benötigen wir das Tensor-Produkt, welches als Resultat einen Spaltenvektor liefert, dessen Einträge die Produkte der jeweiligen Komponenten der Zustände darstellen, wobei die Zahl der Zustände gerade der Zahl der Dualdarstellung entsprechen und die Zahl i im Dezimalsystem gerade an der i-ten von 2^i Stelle des Vektors durch eine 1 dargestellt wird (gezählt wird von oben, beginnend mit 0).

Beispiel:

$$|5\rangle \;=\; |101\rangle \;=\; |1\rangle\,|0\rangle\,|1\rangle \;=\; \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \;=\; (00000100)^T$$

Zuletzt benötigen wir noch die 'Modulo 2 Summe' oder auch 'exklusives oder':

$$1 \oplus 0 \;=\; 0 \oplus 1 \;=\; 1 \qquad 0 \oplus 0 \;=\; 1 \oplus 1 \;=\; 0$$

1.2 Operatoren

Um irgend etwas mit den Zuständen sinnvoll machen zu können, benötigen wir Operatoren, die auf definierte weise auf diese Einwirken, so dass wir mit ihnen Algorithmen zur Berechnung von Funktionen erstellen können.
Beginnen möchte ich mit dem NOT Operator $\mathbf{X}$, der sowohl bei klassischen Computern, als auch bei Algorithmen von Quantencomputern eine wichtige Rolle spielt.
Seine Matrixdarstellung ergibt sich zu

$$\mathbf{X} \;=\; \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

. Wendet man ihn auf einen Zustand an, so invertiert er diesen:

$$\mathbf{X}\,|0\rangle \;=\; |1\rangle \qquad\qquad \mathbf{X}\,|0\rangle \;=\; |1\rangle \qquad\qquad \mathbf{X}^2 \;=\; \mathbf{1}$$

wobei die letzte Relation, zusammen mit der Tatsache, dass $\mathbf{X}^T = \mathbf{X}$ gilt, bedeutet, dass $\mathbf{X}$ unitär ist.
Der NOT-Operator ist ein 1-Qubit Operator. Will man ihn beispielsweise auf ein 2-Qubit-System anwenden, aber nur das erste negieren, so verwendet man ihn wie folgt

$$(\mathbf{X} \otimes \mathbf{1})\,|x\rangle\,|y\rangle \;=\; (\mathbf{X}\,|x\rangle)\,|y\rangle$$

Die Anwendung auf 2-Qubit-Systeme ist ausreichend um die meisten Berechnungen durchzuführen. Die eben beschriebene Erweiterung auf 2 Qubits ist aber auf beliebig viele erweiterbar.
Der erste zwei-Qubit-Operator ist der cNOT ('controlled' NOT) $\mathbf{C}_{ij}$ Operator. Er reagiert auf ein kontroll-Qubit, welches unverändert bleibt und wirkt je nach dessen Wert auf ein Ziel-Qubit:

$$\begin{aligned} \mathbf{C}_{10}\;|x\rangle\,|y\rangle \;&=\; |x\rangle\,|y \oplus x\rangle \\ \mathbf{C}_{01}\;|x\rangle\,|y\rangle \;&=\; |x \oplus y\rangle\,|y\rangle \end{aligned}$$

und deren Matrixdarstellungen sind

$$\mathbf{C}_{10} \;=\; \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad\qquad \mathbf{C}_{10} \;=\; \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Ein weiterer, auch in der klassischen Programmierung bekannter Operator ist der SWAP-Operator, der zwei Zustände austauscht:

$$\begin{aligned} \mathbf{S}_{10}\ |x\rangle\,|y\rangle &= |y\rangle\,|x\rangle \\ \mathbf{S}_{01}\ |x\rangle\,|y\rangle &= |y\rangle\,|x\rangle \end{aligned}$$

und deren Matrixdarstellungen sind

$$\mathbf{S}_{10} = \mathbf{S}_{01} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ein weiterer Operator ist der **Z**- Operator, der, im Gegensatz zu den bisherigen Operatoren, keine klassische Anwendung kennt.
Er wirkt auf alle bis auf den $|11\rangle$ Zustand als Einheitsoperator.
Die dazugehörige Matrixdarstellung lautet:

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Betrachtet man nun den 1-Qubit **Z**- Operator

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

so fällt auf, dass wenn wir einen weiteren Operator

$$\mathbf{Y} = i\mathbf{XZ} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

einführen, so erhalten wir gerade mit **X** die drei Paulimatrizen!
Dies ist aber nicht weiter verwunderlich, denn wir benutzen genau denselben Formalismus zur Beschreibung von Spins, wie hier zur Beschreibung unserer Zustände.
Angemerkt sei noch, dass man mit Hilfe der Pauli-Matrizen $\vec{\sigma} = \begin{pmatrix} \sigma_{\mathbf{x}} \\ \sigma_y \\ \sigma_z \end{pmatrix}$ den SWAP-Operator darstellen kann:

$$\mathbf{S}_{ij} = \frac{1}{2}\left(\mathbf{1} + \vec{\sigma}^{\mathbf{i}} \cdot \vec{\sigma}^{\mathbf{j}}\right)$$

Ebenfalls lässt sich jeder beliebige unitäre Operator **A** darstellen:

$$\mathbf{A} \;=\; a_0\,\mathbf{1} \;+\; \vec{a}\cdot\vec{\sigma}$$

wobei a_0 und die drei Komponenten von $\vec{a}$ reale Zahlen sind.
Nun kommen wir noch zum letzten und wichtigsten Operator, dem Hadamard- Operator:

$$\mathbf{H} \;=\; \frac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z}) \;=\; \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Er wirkt auf die Zustände wie folgt:

$$\begin{aligned} \mathbf{H}\;|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ \mathbf{H}\;|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

Wendet man ihn nun zweimal hintereinander an, so entsteht der Einheitsoperator:

$$\begin{aligned} \mathbf{HH} &= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} 1+1 & 1-1 \\ 1-1 & 1+1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

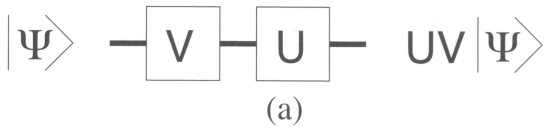
und schließen wir den **Z**- Operator in zwei Hadamard-Operatoren, so erhalten wir den NOT-Operator:

$$\begin{aligned} \mathbf{HZH} &= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix} \end{aligned}$$

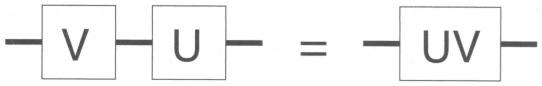
Ihn werden wir in einem späteren Abschnitt benötigen um Zuständig zu präparieren, denn wie wir noch sehen werden, ist dies essentiell für die Funktionsfähigkeit eines Quantencomputers!
Ihnen ist vielleicht aufgefallen, dass bis jetzt alle Operatoren unitär waren ($\mathbf{uu}^\dagger = \mathbf{u}^\dagger\mathbf{u} = \mathbf{1}$)? Dies ist auch eine Forderung, denn so kann man durch Anwenden einer Inverse Operation den Ursprungszustand wieder herstellen, wodurch der Prozess reversibel wird!

1.3 Circuit- Diagramme

Nun kommen wir noch zum letzten Unterabschnitt des ersten Abschnitts, den Circuit- Diagrammen.

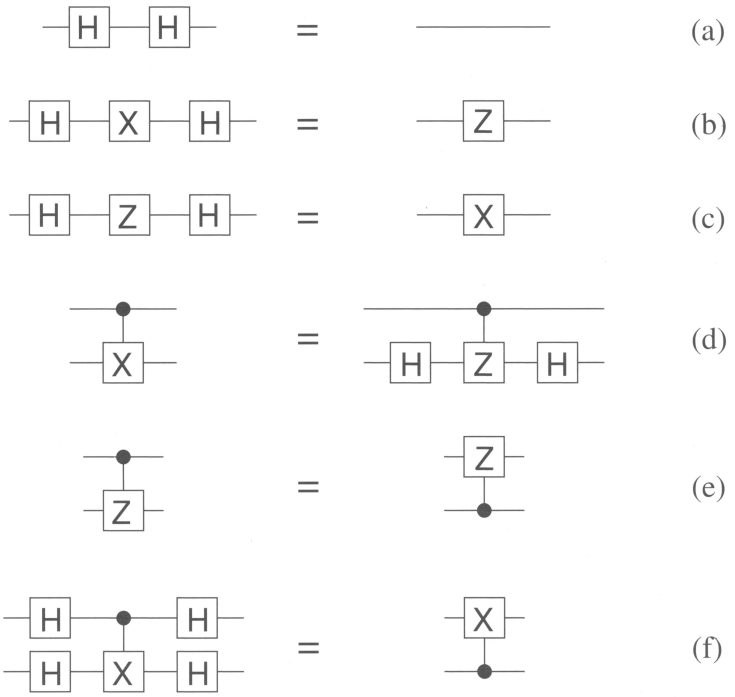


(a)



(b)

Diese dienen allgemein der Veranschaulichung von (zeitabhängigen-) Prozessabläufen, in unserem Fall, von der Anwendung von Operatoren auf einen gegebenen Zustand. Hierbei steht der Ausgangszustand links, auf diesen werden die Operatoren in der Reihenfolge von links nach rechts angewendet und man endet im rechten Zustand. Dabei ist zu beachten, dass in der mathematischen Schreibweise die Operatoren von rechts nach links auf den (Ket-) Zustand wirken, weshalb sie in diesem gegebenen Beispiel am Ende in umgekehrter Reihenfolge da stehen, was durch die untere Darstellung nochmals verdeutlicht werden soll.
Nun nehmen wir noch ein paar explizitere Beispiele:



In (a) erkennt man die schon gezeigte Relation, dass wenn man den Hadamard-Operator zweimal hintereinander anwendet, man beim Einheitsoperator landet, der nicht dargestellt wird (oder nur als durchgezogene Linie).
Auch die Relation unter Punkt (c) haben wir schon gesehen und wenn wir rechts und links jeweils noch einen Hadamard-Operator beifügen, so erhalten wir mit der Relation aus (a) die Relation (b).
Die Relation (d) möchte ich nun noch kurz ausführen:

$$
\begin{aligned}
(\mathbf{1} \otimes \mathbf{H})\,\mathbf{Z}_2\,(\mathbf{1} \otimes \mathbf{H}) &= \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}
\end{aligned}
$$

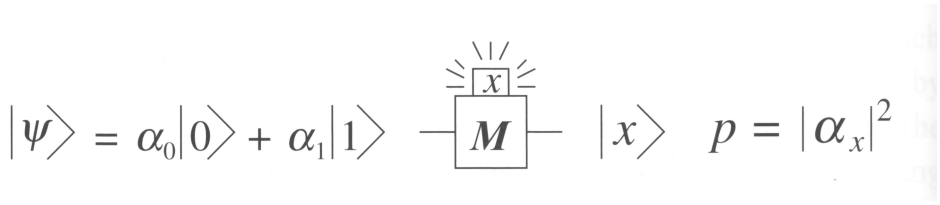
Hierbei meine ich mit dem $\mathbf{Z}_2$ den 'controlled' $\mathbf{Z}$, also den 2-Qubit Operator.
Wie man erkennt, kommt der $\mathbf{C}_{10}$ Operator heraus.
Auch die Relation (e) haben wir schon gezeigt, denn da der $\mathbf{Z}$- Operator nur auf den $|11\rangle$ - Zustand wirkt, ist es egal, welches das Kontroll- und welches das Ziel-Qubit ist!
Die letzte Relation ergibt sich nun aus der Hintereinanderausführung der Relationen (b), (e) und (c).

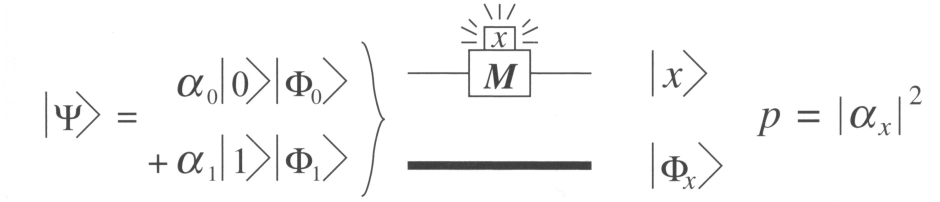
2 Die Born Regel

Nun wollen wir zum zweiten Abschnitt kommen, der Born- und der verallgemeinerten Born-Regel:
Während in klassischen Systemen stets jeder Zustand Abgelesen werden kann, kann man quantenmechanische Zustände nur messen! Dies hat den Kollaps des Zustandes, der zuvor noch als Supperposition von Zuständen

$$|\Psi\rangle = \sum_i \alpha_i |i\rangle$$

mit den Wahrscheinlichkeitsamplituden α_i vorgelegen hat, zu einem festen Wert, der mit der Wahrscheinlichkeit $|\alpha_i|^2$ gemessen wird, zur Folge. Dieser Vorgang ist irreversibel.
Dies wir auch nochmals durch das folgende Circuit-Diagramm veranschaulicht, wobei das M mit 'strahlenden x darüber', den Messprozess darstellt. Während das erste Diagramm noch





die eben beschriebene Bornregel veranschaulicht, geht das zweite Diagramm weiter:
Liegt ein System von mehreren Qubits in einem allgemein verschränkten Zustand $|\Psi\rangle$ vor, so wird die Verschränkung durch die Messung eines einzelnen Qubits (oder eines Teiles der Qubits) die Gesamtverschränkung aufgehoben, wobei man einen festen Wert für das gemessene Qubit und einen neuen Zustand für den Rest erhält.
Hierbei ist anzumerken, dass Untersysteme von Qubits nur dann einen eigenen Zustand besitzen wenn sie nicht mit anderen Untersystemen/Qubits verschränkt sind. Hätte das gemessene Untersystem einen eigenen Zustand, also wäre er nicht mit weiteren Qubits verschränkt, so würde sich nichts an den restlichen durch die Messung ändern.
Da wir nun aber Teilsysteme Messen wollen, ohne dabei ungewollt andere zu verändern, müssen wir diese Präparieren. Als Konvention wird deshalb der Anfangszustand als

$$|0\rangle_n = |0\rangle|0\rangle \cdots |0\rangle \qquad \text{(n- Qubits)} \tag{1}$$

Zustand präpariert. Dies ist dann auch ein wohl definierter Anfangsstartpunkt zur Entwicklung effektiver Algorithmen, wie wir einige einfache im folgenden Abschnitt besprechen wollen.

3 Anwendungen

Nun wollen wir zu den Anwendungen kommen. In diesem Abschnitt will ich nach einem allgemein Teil einige einfache Problemstellungen und deren Lösungen, bzw. deren Anwendungen erörtern.

3.1 allgemeines

Bevor wir zu den Problemstellungen kommen, sollten wir noch ein paar allgemeine Punkte betrachten, die bei der Entwicklung von Algorithmen von Bedeutung sein werden.
Beachten müssen wir, dass wir zwei Register haben, ein Input- und ein Output- Register, wobei wir am Ende beide Werte wissen müssen (dies kennen wir auch schon von klassischen Systemen, wobei hier jedoch der Input stets bekannt ist und deshalb nicht extra ausgelesen werden muss).
Um dies zu erreichen präparieren wir die Zustände mit Hadamard- Operatoren:

$$(\mathbf{H}\otimes\mathbf{H})\,(|0\rangle\otimes|0\rangle) \;=\; \frac{1}{2}\,(|0\rangle_2 \;+\; |1\rangle_2 \;+\; |2\rangle_2 \;+\; |3\rangle_2)$$

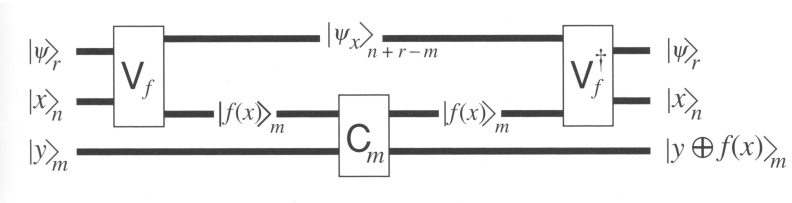
Wenden wir nun einen Funktionsoperator $\mathbf{U}_f$ darauf an, der die Funktion $f(x)$ berechnen soll und deren Wert in den Output schreibt, ohne den Input zu verändern, so erhalten wir

$$\mathbf{U}_f\,(\mathbf{H}\otimes\mathbf{H})\,(|0\rangle\otimes|0\rangle) \;=\; \frac{1}{2^{\frac{n}{2}}}\sum_{0\le x\le 2^n} |x\rangle_n\,|f(x)\rangle_m$$

Wer sich diese Formel nun genauer betrachtet, der stellt fest, dass man mit einem Funktionsaufruf 2^n verschiedene Funktionswerte ermitteln könnte. Dies hat den Begriff Quantenparallelismus geprägt. Doch wenn sie sich noch an die Born'sche Regel erinnern, dann wird Ihnen wohl das Problem dabei schon aufgefallen sein: Was auch immer unseren Zustand beschreibt, messen können wir stets nur einen Wert.
Als möglichen Ausweg könnte man nun auf das Kopieren des Zustandes kommen (sofern Kopieren schneller ist als neu berechnen). Doch dies widerspricht dem 'no-cloning-theorem', was anschaulich klar ist, denn man kann nur kopieren, was man auch kennt!
Als weiteres Problem kommt nun hinzu, dass wir nicht nur unsere $n + m$ Qubits unserer Register haben, sondern r weitere, die auf andere Weise in unser System eingreifen, beispielsweise die zur Messung notwendig sind.
Diese können nun aber mit unseren anderen Qubits verschränken:

$$\mathbf{W}_f\,|\Psi\rangle_{n+m+r} \;=\; |x\rangle_n\,|y\oplus f(x)\rangle_m\,|\Phi\rangle_r$$

Einen Ausweg hierfür bietet wieder eine gute Präparation:



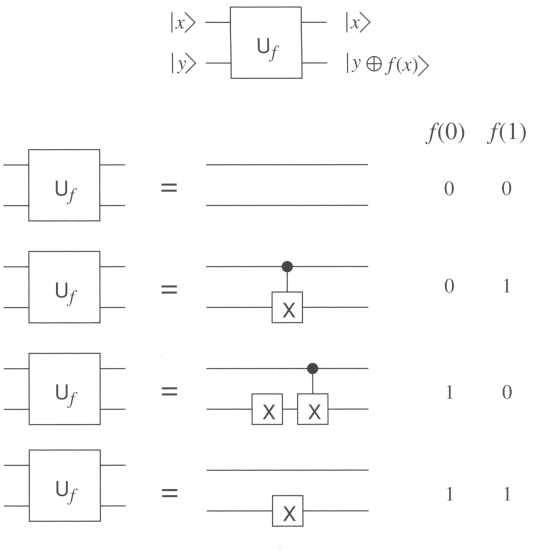
Nehmen wir nun unsere vorpräparierten Zustände und lassen einen unitären Operator $\mathbf{V}_f$, der uns die Funktion berechnen soll, auf den Input und die r zusätzlichen Qubits wirken, so dass wir die Funktionswerte im Input erhalten. Dabei muss gelten:

$$\mathbf{W}_f = \mathbf{V}_f^\dagger \mathbf{C}_m \mathbf{V}_f$$

Nun wenden wir m- mal den cNOT- Operator auf den Input als Kontrollbit und den Output als Zielbit an, wodurch wir nun das Ergebnis unserer Berechnung in den Output verschieben. Um nun auch die Werte zu erhalten, an denen die Funktion berechnet worden ist, müssen wir den inversen V_f - Operator anwenden, wodurch wir wieder unseren unverschränkte Anfangszustand im Input und den zusätzlich Qubits erhalten.

3.2 Deutsch's Problem

Nun wollen wir uns einem konkreten Problem annehmen.
Ein Herr Deutsch fragte sich, was man wohl von den Resultaten einer 'Black-box' lernen könne, in die man nicht hinein sehen kann, also bei der man nicht weiß, welche Operation ausgeführt wird.
Die Problemstellung ist im folgenden Circuit-Diagramm dargestellt.



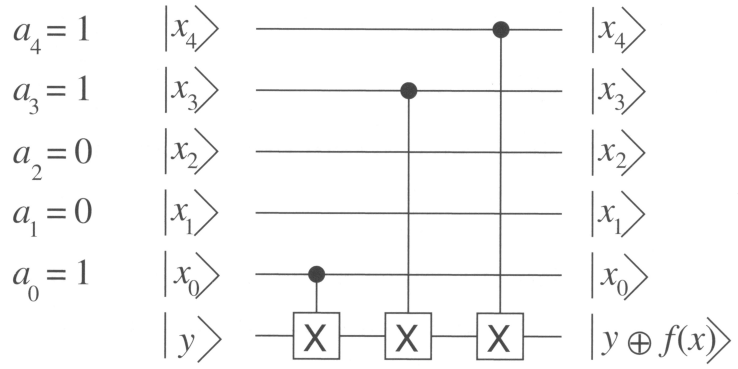
Im speziellen stellte er sich die Frage, was man über die Konstanz der Funktion $f(x)$ aussagen kann, wenn man nur einen Funktionswert berechnet. Im klassischen Fall könnten wir mit nur einem Wert keine aussage treffen.
Wenn man sich das Diagramm betrachtet, so stellt man fest, dass man auch keine Aussage treffen kann.
Doch wenn man den Zustand zuvor präpariert, so erhält man doch eine Lösung dieses Problems:

$$(\mathbf{H}\otimes\mathbf{H})\,\mathbf{U}_f\,(\mathbf{H}\otimes\mathbf{H})\,(\mathbf{X}\otimes\mathbf{X})\,(|0\rangle\,|0\rangle) \;=\; \left\{ \begin{array}{l} |1\rangle\left(|f(0)\rangle \;-\; \left|\tilde{f}(0)\right\rangle\right)\;,\; f(0) \;=\; f(1) \\ |0\rangle\left(|f(0)\rangle \;-\; \left|\tilde{f}(0)\right\rangle\right)\;,\; f(0) \;=\; f(1) \end{array} \right\}$$

wobei $\left|\tilde{f}(x)\right\rangle \;=\; \mathbf{U}_f\,|x\rangle\,|1\rangle$ ist.
Hier sieht man nun, dass man zwar keine Aussage teffen kann, wenn man sich das Outputregister betrachtet, denn die sind in beiden fällten gleich, aber das Inputregister unterschiedlich ist, je nachdem , ob die beiden Funktionswerte gleich ($|1\rangle$) oder unterschiedlich ($|0\rangle$) sind! Dies ist nun das erste Beispiel, bei dem man sieht, dass man mit einer geschickten Präparation bei Quantencomputer Lösungen erhalten kann, wo klassische Computer keine Aussage treffen können.

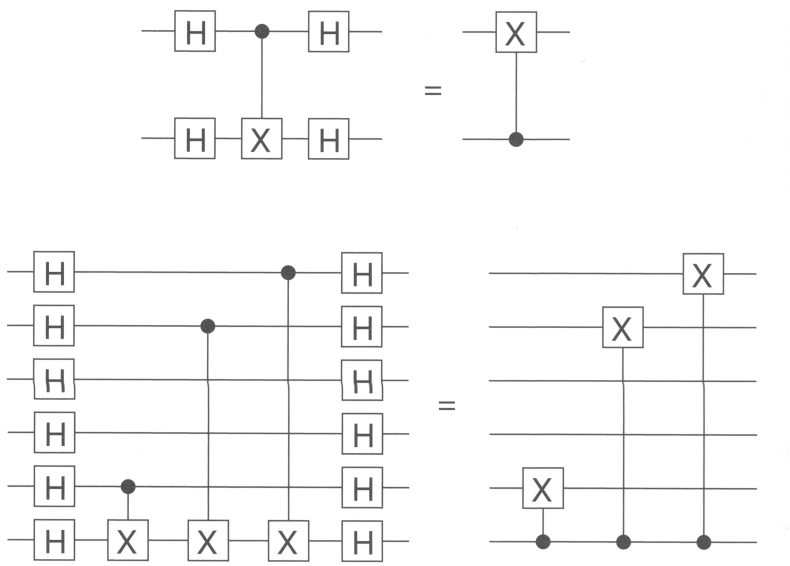
3.3 Bernstein-Problem

Nun kommen wir zu einer Problemstellung, an dessen Ende wir schon etwas mehr Vorteile des Quantencomputers gegenüber klassischer Computer erkennen werden.
Die Aufgabe, die Herr Bernstein sich gestellt hat war folgende:
Gegeben ist eine Funktion $f(x) \;=\; a\cdot x$, wobei das a gesucht ist. Hier bedeutet der Punkt das modulo 2 Skalarprodukt.
Kurzes Beispiel:
$a \;=\; 25 \quad x_0 \;=\; 5$:
$f(x_0) \;=\; (11001)\cdot(00101)^T \;=\; 1$
Betrachten wir nun das Circuit - Diagramm des Problems:



Hier erkennen wir, dass überall, wo unser a eine 1 hat, der cNOT- Operator im Output den Wert umdreht.

Da wir so nicht sehr viel sehen, wenden wir unsere Hadamard- Operatoren an zwei mal an (Erinnerung: $\mathbf{HH} = \mathbf{1}$).



Was wir nun erhalten ist, dass an den Stellen, an denen $a_i = 1$ ist, der cNOT- Operator für eine Invertierung des Inputs sorgt!
Haben wir unser Inputregister nach Konvention auf $|0\rangle$ Präpariert, so steht nun das gewünschte Ergebnis nach nur einem Funktionsaufruf im Input! Dies bedeutet einen Rechenvorteil von n auf 1.
Mathematisch sieht das Ganze wie folgt aus:

$$\mathbf{H}^{\otimes(n+1)}\mathbf{U}_f\mathbf{H}^{\otimes(n+1)}\,|0\rangle_n\,|1\rangle_1 \;=\; |a\rangle_n\,|1\rangle_1$$

3.4 Simon's Problem

Zum Schluss wollen wir mit dem Simon's Problem noch kurz eine Anwendung zeigen, dessen Rechenvorteil gegenüber klassischen Computern noch deutlicher sichtbar wird.
Beim Simon's Problem geht es darum, dass eine periodische Funktion ($f(x \oplus a) = f(x)$) gegeben ist, wobei die Periode a gesucht ist.
Ein klassischer Computer benötigt für eine 100 Bit Verschlüsselung etwa $2^{\frac{n}{2}} = 2^{50} \approx 10^{15}$ Funktionsaufrufe, ein Quantencomputer schafft dies mit etwa $n + \Delta n \approx 120$ Funktionsaufrufen.
Nun, wie funktioniert das:
Wenden wir wieder unsere Funktion $\mathbf{U}_f$ auf unseren Präparierten Zustand an, dann erhalten wir:

$$\mathbf{U}_f \left(\mathbf{H}^{\otimes n} \otimes \mathbf{1}\right) |0\rangle |0\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle$$

nach Born erhalten wir nun im Input $\frac{1}{\sqrt{2}} (|x_0\rangle + |x_0 \oplus a\rangle)$
wenden wir nun hierauf wieder die Hadamard- Operatoren an, so erhalten wir:

$$\begin{aligned} \mathbf{H}^{\otimes n} \frac{1}{\sqrt{2}} (|x_0\rangle + |x_0 \oplus a\rangle) &= \frac{1}{2^{(n+1)/2}} \sum_{y=0}^{2^n-1} \left((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus a) \cdot y}\right) |y\rangle \\ &= \frac{1}{2^{(n+1)/2}} \sum_{a \cdot y = 0} (-1)^{x_0 \cdot y} |y\rangle \end{aligned}$$

Da wir nun die y-ons Messen können und wir über die Bedingung $y \cdot a = 0$ direkt auf a schließen können, erhalten wir wirklich nach n Aufrufen mit einer kleinen Unsicherheit (doppelte Funktionsauswertungen) unsere Lösung.
Zur Veranschaulichung noch ein kleines Beispiel:
Unser a sei $5 = 101$, wir messen nun mit der Bedingung $y \cdot a = 0$ y:
1. Messung: $y = 010$
$\rightarrow$ mögliche Werte für a: $\left\{ \begin{array}{ccc} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{array} \right\}$
2. Messung: $y = 101$
$\rightarrow$ $a \cdot y = \{1\ ,\ 1\ ,\ 0\}$ für die 3 zuvor übrig gebliebenen Werte für a bleibt nur letzte übrig
$\rightarrow$ $a = 101$

4 Zusammenfassung

Wir haben nun nach einiger Einführung gesehen, dass es keinen Quantenparallelismus gibt. Beim letzten Teil haben wir auch gesehen, dass Quantencomputer bei manchen Problemstellungen doch deutliche Rechenvorteile gegenüber klassischen Computern haben (Simon: exponentiell gegen linear.)

5 Quellenangabe

Dieser Vortrag stellt eine Zusammenfassung der Kapitel 1 + 2 von N. David Mermin, *Quantum Computer Science, An Introduction*, Cambridge University Press 2007