AℓZAGK Seminar: Quantencomputer
3.Vortrag
QBits und einfache Gatter

Ingolf Meyer

8. Mai 2001

3.1 Einleitung

Um einen möglichen Quantencomputer zu realisieren, ist es nötig, mehr als nur ein QBit zu betrachten. Es soll in diesem Vortrag also darum gehen, die mathematische Beschreibung der QBits fortzusetzen und auf Systeme von mehreren QBits auszudehnen. Es wird dabei insbesondere das Tensorprodukt eingeführt werden, das die mathematische Beschreibung von Systemen von Quantenzuständen ermöglicht. Am Ende sollen dabei noch einige einfache Gatter besprochen werden.

3.2 Systeme von QBits und das Tensorprodukt

Definition 3.1 (QBit)
Ein QBit ist ein zweidimensionaler Hilbertraum H mit der Standardbasis, die als $|0\rangle = e_1$ und $|1\rangle = e_2$ geschrieben wird. Der Zustand oder Wert eines QBits ist ein Einheitsvektor in H.

Wir können nun für ein QBit eine beispielhafte Observable angeben, die als Eigenvektoren die Basisvektoren $|0\rangle, |1\rangle$ und als Messwerte $\frac{1}{2}$ bzw. -$\frac{1}{2}$ hat. Die Matrix hierfür ist:

$$A = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Wir wollen hier nocheinmal kurz das System des Messens eines QBits erläutern. Dieses System ist durch die Postulate der Quantenmechanik gegeben und kann daher hier nur als Verfahren beschrieben werden. Nehmen wir an, dass sich das QBit im Zustand ψ befindet, dann bedeutet eine Messung des Messwerts a einer Observablen A, dass das System sich danach im Zustand

$$\psi\prime = \frac{P_a(\psi)}{|P_a(\psi)|^2}$$

befindet, wobei P_a der orthogonalen Projektor auf den Eigenraum von A zum Eigenwert a ist. Mit anderen Worten wird das System durch die Messung orthogonal in den Eigenraum zu a projeziert und anschließend wieder normiert. Interessiert man sich nun für die Wahrscheinlichkeit, den Messwert a bei einer Messung zu erhalten, so ist diese durch

$$\pi_a := \langle\psi|P_a(\psi)\rangle$$

gegeben.

Nun stellt sich natürlich die Frage, wie es sich mit Systemen von mehreren QBits verhält. Ein klassisches Bit kann als F_2 verstanden werden und ein System von n klassischen Bits ist dann das kartesische Produkt von n Exemplaren des F_2, d.h. F_2^n. Bei QBits müssen wir stattdessen das Tensorprodukt verwenden. Wir wollen es hier allgemein für Vektorräume einführen.

Satz 3.2 (Tensorprodukt)
Seien V und W zwei K-Vektorräume, dann gibt es einen K-Vektorraum $V \otimes W$ und eine bilineare Abbildung $\otimes : V \times W \to V \otimes W$, die folgende universelle Eigenschaft hat:

Zu jeder bilinearen Abbildung $\beta : V \times W \to U$, wobei U ein beliebiger K-Vektorraum ist, gibt es eine eindeutig bestimmte lineare Abbildung $\lambda_\beta : V \otimes W \to U$, so dass $\lambda_\beta \circ \otimes = \beta$.

Falls V und W endlich sind, gilt (dim V)·(dim W) = dim$V \otimes W$.

Beweis:

Sei $\{v_i\}_{i \in I}$ eine Basis von V und $\{w_j\}_{j \in J}$ eine Basis von W, dann schreiben wir den Vektorraum $V \otimes W$ als Vektorraum der formalen, endlichen Linearkombinationen von Elementen, die wir mit bezeichnen $v_i \otimes w_j$ *wollen*. Die Abbildung $\otimes$ bildet dann (v_i, w_j) auf $v_i \otimes w_j$ ab und ist eindeutig bestimmt, da $\otimes$ als bilineare Abbildung durch seine Werte auf den Paaren von Basisvektoren eindeutig bestimmt ist. Falls dim V = n und dim W= m ist, gibt es $n \cdot m$ Ausdrücke der Form $v_i \otimes w_j$ und $V \otimes W$ hat die Dimension $n \cdot m$.

Nun ist zu zeigen, dass der so konstruierte Vektorraum $V \otimes W$ mit der Abbildung $\otimes$ die geforderte universelle Eigenschaft besitzt. Dazu ist λ_β anzugeben, das wir wie folgt definieren:

$$\lambda_\beta(v_i \otimes w_j) = \beta(v_i, w_j) \qquad \text{für alle i und j.}$$

Wir haben λ_β nun auf einer Basis von $V \otimes W$ festgelegt und damit ist λ_β als lineare Abbildung eindeutig bestimmt. Da $\otimes$ bilinear ist, ist damit auch $\lambda_\beta \circ \otimes$ bilinear. Schließlich bleibt noch festzustellen, dass $\lambda_\beta \circ \otimes$ gleich β ist, da beide bilinear sind und auf den Basisvektorpaaren (v_i, w_j) gleich sind. Ω

Bemerkungen

- Nicht alle Elemente von $V \otimes W$ lassen sich in der Form $v \otimes w$ darstellen, wobei $v \in V$ und $w \in W$, sondern im allgemeinen läßt sich ein Element $x \in V \otimes W$ nur als Linearkombination von Elementen der Form $v \otimes w$ darstellen.
- Die Bilinearität von $\otimes$ drückt sich so aus:

$$\begin{aligned} v \otimes w + \tilde{v} \otimes w &= (v + \tilde{v}) \otimes w \\ v \otimes w + v \otimes \tilde{w} &= v \otimes (w + \tilde{w}) \\ (rv) \otimes w = r(v \otimes w) &= v \otimes (rw) \end{aligned}$$

In Bezug auf Quantencomputer wird uns auch interessieren wie wir lineare Abbildungen von V und W zu einer linearen Abbildung auf $V \otimes W$ zusammenfassen können, die dabei noch einige gewünschte Eigenschaften von V und W erhält. Dazu hilft folgendes Lemma:

Lemma 3.3 (Tensorprodukt von linearen Abbildungen)
Seien V und W zwei K-Vektorräume mit Tensorprodukt $V \otimes W$ und Basen $v_1, \ldots, v_n$ und $w_1, \ldots, w_m$. Seien ferner $f : V \to V$ und $g : W \to W$ zwei lineare Abbildungen mit Matrizen A=$(a_{k,l})$ bzw. B=$(b_{r,s})$, dann wird $f \otimes g : V \otimes W \to V \otimes W$ als lineare Abbildung durch $(f \otimes g)(v_i \otimes w_j) = f(v_i) \otimes g(w_j)$ definiert. Die Matrix $A \otimes B$ von $f \circ g$ bezüglich der $v_i \otimes v_j$ ist durch $(a_{i,j} \cdot b_{r,s})_{ir,js}$ gegeben.

Beweis:

Die Abbildung $f \times g : V \times W \to V \times W$ ist linear und die Abbildung $h = \otimes \circ (f \times g)$ bilinear, was man unter Benutzung der Rechenregeln nachrechnet. Aufgrund der universellen Eigenschaft läßt sich eindeutig die lineare Abbildung $f \otimes g$ angeben, für die gilt: $h = (f \otimes g) \circ \otimes$. Diese ist gerade die gesuchte.

Jetzt ist noch die Matrizendarstellung zu zeigen. Dazu betrachten wir die folgenden Gleichungen:

$$\begin{aligned} f(v_i) &= a_{1,i}v_1 + \ldots + a_{n,i}v_n \\ g(w_j) &= b_{1,j}w_1 + \ldots + b_{m,j}w_m. \end{aligned}$$

Damit haben wir:

$$\begin{aligned} & (f \otimes g)(v_i \otimes w_j) = f(v_i) \otimes g(w_j) \\ &= (a_{1,i}v_1 + \ldots + a_{n,1}v_n) \otimes (b_{1,j}w_1 + \ldots + b_{m,1}w_m) \\ &= \begin{array}{c} a_{1,i}b_{1,j}v_1 \otimes w_1 + \ldots + a_{n,i}b_{1,j}v_n \otimes w_1 + \\ \vdots \\ a_{1,i}b_{m,j}v_1 \otimes w_m + \ldots + a_{n,i}b_{m,j}v_n \otimes w_m \end{array} \\ &= \sum_r \sum_k a_{r,i}b_{k,j}v_r \otimes w_k. \end{aligned}$$

Nun muss man nur noch die Bilder der Basisvektoren in die Spalten schreiben und damit ist also die Matrix $A \otimes B$ gegeben durch $(a_{i,j} \cdot b_{r,s})_{ir,js}$. Ω

Man kann sich vielleicht die obige Formel für die Matrixelemente leichter merken, wenn man sich folgendes Schema merkt:

$$A \otimes B = \begin{bmatrix} a_{1,1}[B] & \ldots & a_{1,n}[B] \\ \vdots & \ddots & \vdots \\ a_{n,1}[B] & \ldots & a_{n,n}[B] \end{bmatrix}$$

Die obigen Konstruktionen kann man analog auch für Tensorprodukte von n Vektorräumen $V_1 \otimes \ldots \otimes V_n$ durchführen. Wir wollen dies hier aber nicht ausführen, sondern werden diese nur als Iterationen von Tensorprodukten zweier Vektorräume auffassen, die man beliebig „klammern“darf.

Damit sind nun endlich die nötigen Grundlagen geschaffen, Systeme von n QBit zu betrachten.

Definition 3.4 (System von QBits)
Ein System H von n QBits $H_1, \ldots, H_n$ ist definiert als Tensorprodukt der einzelnen QBits,d.h.

$$H = H_1 \otimes \ldots \otimes H_n.$$

Als positiv definite Sequilinearform auf H führen wir ein:

$$\langle u_1 \otimes \ldots \otimes u_n, v_1 \otimes \ldots \otimes v_n \rangle := \langle u_1, v_1 \rangle \ldots \langle u_n, v_n \rangle .$$

Da diese auf den Basisvektoren von H bestimmt ist, ist sie damit auf ganz H eindeutig festgelegt. Die Vollständigkeit folgt aus der Vollständigkeit der Einzelräume. Für die Basisvektoren von H wird häufig die folgende abkürzende Schreibweise gebraucht, wobei $|i_j >$ die Basisvektoren von H_j meint:

$$|i_1 \ldots i_n\rangle := |i_1\rangle \otimes \ldots \otimes |i_n\rangle.$$

Insgesamt gibt es also 2^n solche Vektoren, wobei jeder Index i_k nur die Werte 0 und 1 annehmen kann. Damit hat man also bei n QBits als Vektoren die binäre Darstellung der Zahlen $0, 1, \ldots, 2^n - 1$. Der Vektor $|101\rangle$ kann z.B. als Zahl 9 in der binären Darstellung 101 aufgefasst werden.

3.3 Nichtlokalität (EPR)

Eine erstaunliche Erscheinung der Quantenmechanik ist die Nichlokalität, d.h. für uns die Messung eines QBits kann in einem System von mehreren QBits auch andere QBits beeinflussen. Dies wollen wir an einem Beispiel erläutern

Betrachten wir einen Zustand

$$\psi = (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_1b_2|11\rangle$$

eines 2-QBit-Systems. Wird nun der Zustand des ersten QBits gemessen und sei dieser $|0\rangle$, dann hat ψ die Gestalt:

$$\psi = |0\rangle \otimes (a_2|0\rangle + b_2|1\rangle).$$

Die Messung des ersten QBits hat in diesem Fall also keinen Einfluss auf das zweite QBit. Anders sieht es aus, wenn wir einen Zustand χ der folgenden Form betrachten:

$$\chi = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle.$$

Die Wahrscheinlichkeit das erste QBits im Zustand 1 zu messen, wenn das zweite bereits als 0 gemessen wurde, ist hier 1. Die Wahrscheinlichkeit das das erste QBit im Zustand eins zu messen, wenn das zweite QBit noch nicht gemessen wurde beträgt aber nur $\frac{1}{2}$. Die Messung der QBits ist also nicht unabhängig. Man sagt auch der Zustand χ ist *verschränkt*. Es wird, wie wir später noch sehen werden, eine entscheidende Rolle in den Algorithmen spielen, eine große Anzahl QBits zu verschränken.

3.4 Unitäre Transformationen auf QBits

Nun gibt es zwei Möglichkeiten, auf QBits und Systeme davon Einfluss zu nehmen. Die eine ist das Messen ihres Zustands, was diesen aber unwiederruflich zerstört. Die andere Möglichkeit ist das Anwenden von unitären Transformationen auf die QBits. Diese gestatten es nämlich, die Wahrscheinlichkeit einzelner Messungen zu beeinflussen, ohne das ganze System zu zerstören. Es handelt sich dabei in gewisserweise um Operationen auf den QBits „ohne Hinsehen". Dass nur unitäre Transformationen dies leisten, hängt damit zusammen, dass sich die Lösungen der Schrödinger-Gleichung unitär in der Zeit entwickeln. Dieser Punkt soll aber hier nicht weiter erläutert werden.

Insbesondere sind unitäre Transformationen aber umkehrbar, d.h. im Gegensatz zu klassischen Computern müssen Quantencomputer bis zur Durchführung von Messungen am System vollständig reversibel rechnen.

Als Beispiel soll auf ein QBit die folgende unitäre Transformation angewendet werden:

$$R = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Man nennt sie die Hadamard-Transformation, sie bildet den Vektor (a, b) auf $\frac{1}{\sqrt{2}}(a + b, a - b)$ ab. Aus dem reinen Zustand $|0\rangle$ wird also unter H der Zustand $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Hier ist nun die Wahrscheinlichkeit den Wert 0 zu messen nur noch

$\frac{1}{2}$. Es ist also möglich durch „erlaubte“ unitäre Transformationen die Wahrscheinlichkeiten für bestimmte Messergebnisse zu ändern.

Eine anderes Beispiel für eine Transformation ist

$$N = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Diese Matrix bewirkt das der Zustand $a|0\rangle + b|1\rangle$ in den Zustand $b|0\rangle + a|1\rangle$ übergeht. Die Transformation N wird auch logisches *NICHT* genannt, weil sie die Basiszustände wie folgt transformiert:

$$\begin{aligned} |0\rangle &\mapsto |1\rangle \\ |1\rangle &\mapsto |0\rangle \end{aligned}$$

Nun ist es nur natürlich, dass man auch Transformationen auf mehrere QBits ausdehnen will. So möchte man zum Beispiel ein logisches Nicht auf das erste von zwei QBits anwenden, während man das zweite unverändert läßt. Diese Transfomation hat die Matrix:

$$N \otimes I = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Wendet man diese Transformation auf den Zustand

$$\psi = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$$

an, so wird aus diesem der Zustand

$$\tilde{\psi} = a_{10}|00\rangle + a_{11}|01\rangle + a_{00}|10\rangle + a_{01}|11\rangle.$$

Strenggenommen hätten wir zuerst prüfen müssen, ob es sich bei $N \otimes I$ um eine unitäre Transformation handelt, was wir nun allgemein nachholen.

Satz 3.5 *Seien $V_1, \ldots, V_n$ endlich dimensionale Hilberträume. Für $i = 1, \ldots, n$ sei U_i eine unitäre Transformation auf V_i, dann ist $U_1 \otimes \ldots \otimes U_n$ eine unitäre Transformation auf $V_1 \otimes \ldots \otimes V_n$.*

Beweis:
Allgemein gilt folgender Zusammenhang

$$(A_1 \otimes \ldots \otimes A_n)(B_1 \otimes \ldots \otimes B_n) = (A_1 B_1 \otimes \ldots \otimes A_n B_n)$$

für lineare Abbildungen A_i, B_i. Nun sieht man, dass

$$(U_1^\dagger \otimes \ldots \otimes U_n^\dagger)(U_1 \otimes \ldots \otimes U_n) = I,$$

also bleibt nur nocht zu zeigen, dass gilt:

$$(U_1^\dagger \otimes \ldots \otimes U_n^\dagger) = (U_1 \otimes \ldots \otimes U_n)^\dagger.$$

Dies folgt durch Ausrechnen mit Hilfe der expliziten Formel wie in 3.1 angegeben und Induktion oder direkt aus der Definition des Skalarprodukts.

Ω

Als letztes Beispiel wollen wir noch eine Transformation angeben, die sich nicht als Tensorprodukt zweier einzelner unitärer Transformationen schreiben läßt. Sie soll als Ergebnis bei einem zwei QBit-System das erste Bit unverändert lassen und in das zweite das Ergebnis der *XOR*-Operation schreiben. Damit müssen die Basiszuständen wie folgt abgebildet werden:

$$\begin{aligned} |00\rangle &\mapsto |00\rangle \\ |01\rangle &\mapsto |10\rangle \\ |10\rangle &\mapsto |11\rangle \\ |11\rangle &\mapsto |10\rangle \end{aligned}$$

Mit anderen Worten gesprochen hat die X genannte Transformation die Matrix:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

.

Den Abschluss diese Vortrags soll ein Satz von Deutsch bilden, der sagt, dass es für die tatsächliche Implementierung beliebiger unitärer Transformationen auf n QBits reicht, beliebige Transformationen auf nur zwei QBits durchführen zu können. Tatsächlich wird dies Resultat nicht gut genug sein, denn es ist sehr unwahrscheinlich, dass man beliebige unitäre Transformationen auf zwei QBits direkt realisieren kann. Es gibt jedoch Sätze, die beweisen, dass es gewissen Matrizen gibt, die als Generatoren in der Menge aller unitären Matrizen dicht liegen. Diese hier aufzuführen, würde jedoch den Rahmen des Vortrags sprengen.

Satz 3.6 (David Deutsch,1985) *Sei V ein Hilbertraum der Dimension n mit beliebiger Basis B. Jede unitäre Matrix U auf V läßt sich als Produkt von* $2n^2 - n$ *unitären Matrizen, die auf Unterräumen operieren, die von zwei Basisvektoren aus B aufgespannt werden.*

Für Quantencomputer bedeutet dies, dass es ausreicht auf je zwei QBits operieren zu können.

Beweis:

Seien $v_1, \ldots, v_n$ eine ON-Basis von Eigenvektoren von U und $\lambda_1, \ldots, \lambda_n$ die dazu gehörenden Eigenwerte, d.h. λ_i ist Eigenwert mit Eigenvektor v_i. Die λ_i müssen also nicht notwendig verschieden sein.

Bezüglich B habe v_1 die Darstellung $(x_1, \ldots, x_n)$. Bildet man nun die unitäre Matrix

$$A_{12} = \left[\begin{array}{c|c} \frac{1}{\sqrt{|x_1|^2+|x_2|^2}} \begin{bmatrix} \bar{x_1} & \bar{x_2} \\ -x_2 & x_1 \end{bmatrix} & 0 \\ \hline 0 & \begin{matrix} I_2 & & 0 \\ & \ddots & \\ 0 & & I_2 \end{matrix} \end{array}\right],$$

so bildet diese den Vektor v_1 auf den Vektor

$$(\sqrt{|x_1|^2 + |x_2|^2}, 0, x_3, \ldots, x_n)$$

ab. Bildet man nun noch n-2 weitere unitäre Matrizen A_{1k}, die analog die erste Komponente x_1 mit der k-ten Komponente x_k verknüpfen, dann ergibt sich:

$$A_{1n} \cdot \ldots \cdot A_{12} v_1 = e_1,$$

wobei e_1 der erste Standardbasisvektor von B ist. Multipliziert man nun mit dem Skalar λ_1 und wendet anschließend die Inversen der A_{1k} rückwärts an, dann wirkt die unitäre Abbildung

$$A_1 = A_{12}^{-1} \cdot \ldots \cdot A_{1n}^{-1} \cdot \begin{bmatrix} \lambda_1 & & & 0 \\ & 1 & & \\ & & \ddots & \\ 0 & & & 1 \end{bmatrix} A_{1n} \cdot \ldots \cdot \cdot A_{12}$$

durch Multiplikation mit λ_1 auf v_1. Alle anderen Basisvektoren $v_2, \ldots, v_n$ bleiben davon unbeeinflußt. A_1 besteht dabei aus 2(n-1)+1 unitären Transformationen, die auf maximal zwei Basisvektoren operieren, und ist natürlich selbst wieder unitär. Analog lassen sich auch für die anderen Basisvektoren unitäre Abbildungen $A_2, \ldots, A_n$, die aus 2n-1 unitären Abbildungen bestehen, die auf nur zweidimensionalen Unterräumen operieren. Führt man diese nun hintereinander aus, dann bekommt man eine unitäre Abbildung, die aus n(2n-1) unitären Abbildungen besteht, die jeweils nur maximal zwei Baisvektoren beeinflussen. Diese stimmt auf allen Basisvektoren mit U überein und ist deshalb gleich U. Ω