Quantum Computing Klausurvorbereitung

Basics

Tensorprodukt

Wenn A eine Matrix der Größe $m \times n$ ist und B eine Matrix der Größe $p \times q$, dann ist das Tensorprodukt $A \otimes B$ eine Matrix der Größe $(mp) \times (nq)$, die wie folgt definiert ist:

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}$$

Beispiel:

Sei $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ und $B = \begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix}$. Dann ist $A \otimes B$:

$$A \otimes B = \begin{pmatrix} 1\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} & 2\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} \\ 3\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} & 4\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{pmatrix}$$

Matrixmultiplikation

Wenn A eine Matrix der Größe $m \times n$ ist und B eine Matrix der Größe $n \times p$, dann ist das Produkt $C = AB$ eine Matrix der Größe $m \times p$, wobei jedes Element c_{ij} von C wie folgt berechnet wird:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Beispiel:

Sei $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ und $B = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}$. Dann ist $C = AB$:

$$C = \begin{pmatrix} 1 \cdot 2 + 2 \cdot 1 & 1 \cdot 0 + 2 \cdot 3 \\ 3 \cdot 2 + 4 \cdot 1 & 3 \cdot 0 + 4 \cdot 3 \end{pmatrix} = \begin{pmatrix} 2 + 2 & 0 + 6 \\ 6 + 4 & 0 + 12 \end{pmatrix} = \begin{pmatrix} 4 & 6 \\ 10 & 12 \end{pmatrix}$$

Good to Know:

$$\mathrm{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \|x\|_2 = \|x\| = \sqrt{\sum_{k=1}^{n} |x_k|^2}$$

$$e^{i\pi} = -1 \qquad e^{i2\pi} = e^0 = 1 \qquad e^{i\frac{3\pi}{2}} = -i$$

$$-i \cdot (-i) = i^2 = -1 \qquad i \cdot (-i) = 1 \qquad e^{i\frac{\pi}{2}} = i$$

$$\|i\| = 1 \qquad e^{i\theta} = cos\theta + isin\theta$$

Der irrationale Teil kann als globale Phase verworfen werden.

Quantum Computing Basics

Ein **Qubit** ist die simpelste Einheit einer Quantum Information. Ein Qubit kann 0,1 oder in einer Superposition von 0 und 1 gleichzeitig sein.

Ein Qubit in **Superposition** ist in der folgenden Form: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, wobei α und β komplexe Nummern sind, sodass $|\alpha|^2 + |\beta|^2 = 1$.

Bra-Ket

Ket $|0\rangle$: Ist der Spaltenvektor in einem komplexen Vector Space (Hilbert Space) $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$

Bra $\langle 0|$: Ist das hermetische konjugierte Ket, also ein Zeilenvektor, welcher die konjugierte Transposition des Ket ist $\langle 0| = \begin{pmatrix}1 & 0\end{pmatrix}$

Inner Product: $\langle 0|0\rangle = \begin{pmatrix}1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = 1$, 0 wenn Vektoren orthogonal zueinander

Outer Product: $|0\rangle\langle 0| = \begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1 & 0\end{pmatrix} = \begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}$

Quantum Gates

Quatum Gates manipulieren Qubits durch unitäre Operationen.

Pauli-X-Gate, Bit Flip: $X = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$

Pauli-Y-Gate, Bit and Phase Flip: $Y = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix}$

Pauli-Z-Gate, Phase Flip: $Z = \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}$

Hadamard Gate, erstellt Superposition: $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$

Phase Gate, fügt Phase auf $|1\rangle$ hinzu: $S = \begin{pmatrix}1 & 0\\0 & i\end{pmatrix}$

CNOT Gate, zweites Qubit wird geflippt, wenn erstes Qubit 1 ist: $CNOT = \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}$

T-Gate, S-Gate mit Phase Shift $\pi/4$: $T = \begin{pmatrix}1 & 0\\0 & e^{i\pi/4}\end{pmatrix}$

Rotation Gates

Rotation-X Gate, rotiert um die X-Achse: $R_x(\theta) = \begin{pmatrix}\cos(\theta/2) & -i\sin(\theta/2)\\-i\sin(\theta/2) & \cos(\theta/2)\end{pmatrix}$

Rotation-Y Gate, rotiert um die Y-Achse: $R_y(\theta) = \begin{pmatrix}\cos(\theta/2) & -\sin(\theta/2)\\\sin(\theta/2) & \cos(\theta/2)\end{pmatrix}$

Rotation-Z Gate, rotiert um die Z-Achse: $R_z(\theta) = \begin{pmatrix}e^{-i\theta/2} & 0\\0 & e^{i\theta/2}\end{pmatrix}$

Gate Identities

$$XX = I \qquad XY = -YX \qquad HXH = Z$$
$$YY = I \qquad YZ = -ZY \qquad HZH = X$$
$$ZZ = I \qquad ZX = -XZ \qquad HYH = -Y$$
$$TT = S \qquad SS = Z \qquad U^\dagger U = UU^\dagger = I$$
$$S^\dagger S^\dagger = Z \qquad H^\dagger = H \qquad XZX = -Z$$
$$ZS^\dagger = S \qquad SZS = I \qquad \frac{X+Z}{\sqrt{2}} = H$$
$$U^{\dagger\dagger} = U \qquad iXZ = Y$$

Measurement

Die Z-Basis (auch computational Basis) besteht aus den States $|0\rangle, |1\rangle$, welche den klassischen binären Werten 0 und 1 entsprechen.

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

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

Beim Messen kollabiert ein State $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ in einen der beiden Basic States $(|0\rangle, |1\rangle)$ mit den Wahrscheinlichkeiten $|\alpha|^2$ und $|\beta|^2$.

Die X-Basis (auch Hadamard Basis) besteht aus den States $|+\rangle, |-\rangle$, welche die Superposition von $|0\rangle, |1\rangle$ sind.

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

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

Ein X-Basis State kann erstellt werden durch das Anwenden von Hadamard auf der Z-Basis, dies geht auch in die andere Richtung gleichermaßen

Beim Messen kollabiert ein State $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ in einen der beiden Basic States $(|+\rangle, |-\rangle)$ mit den Wahrscheinlichkeiten $\left|\frac{\alpha+\beta}{\sqrt{2}}\right|^2$ und $\left|\frac{\alpha-\beta}{\sqrt{2}}\right|^2$.

Messen des $|+\rangle$ States in $|\psi\rangle$ (X-Basis, Wahrscheinlichkeit für den Outcome $|+\rangle$): $\|\langle+|\psi\rangle\|^2$

Nach der Messung kollabiert der State in den Post-Measurement State (dieser ist normalisiert).

Man geht da in der folgenden Abfolge vor (am Beispiel eines 4-Qubit Systems, bei dem wir im 2. Qubit 0 und im dritten 1 messen wollen, Identity für die nicht relevanten Qubits):

1. Projektor konstruieren: $P = I \otimes |0\rangle\langle 0| \otimes |1\rangle\langle 1| \otimes I$
2. Projektor auf den Zustand anwenden (Collect): $|\psi'\rangle = P|\psi\rangle$
3. Wahrscheinlichkeit berechnen: $\||\psi'\rangle\|_2^2$
4. Zustand normalisieren:

$$|\psi_{post}\rangle = \frac{|\psi'\rangle}{\sqrt{\langle\psi'|\psi'\rangle}}$$

Andere Möglichkeit, am Beispiel um 0 im ersten Bit zu messen: $\frac{1}{\sqrt{3}}|+ +\rangle + \sqrt{\frac{2}{3}}|- -\rangle$

1. $\frac{1}{\sqrt{3}}(|+\rangle|+\rangle) + \sqrt{\frac{2}{3}}(|-\rangle|-\rangle)$, wir betrachten nur das erste Bit, daher berechnen wir
2. $\langle 0|\frac{1}{\sqrt{3}}|+\rangle + \langle 0|\sqrt{\frac{2}{3}}|-\rangle$

Propability of Outcome

1. Falls notwendig gate applien
2. Inner product von state und dem outcome den man möchte
3. Wahrscheinlichkeit berechnen

Inner Product am Beispiel – State: $\langle -|\psi\rangle = \left(\frac{1}{\sqrt{2}}(\langle 0| - \langle 1|)\right)(a|0\rangle + b|1\rangle)$

$P(+) = \frac{1}{2}|a+b|^2$

$P(-) = \frac{1}{2}|a-b|^2$

$P(0) = |a|^2$

$P(1) = |b|^2$

Express the resulting state in a computational Basis

Bei einem State $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ nehmen wir die beiden Faktoren α und β und führen eine Matrixmultiplikation mit dem entsprechenden Gate durch.

Beispiel: $T^{2022}\left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle\right)$

$$T^{2022} = \begin{pmatrix} 1 & 0 \\ 0 & \left(e^{i\frac{\pi}{4}}\right)^{2022} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i(2022 \bmod 8)\frac{\pi}{4}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{6}{4}\pi} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$$

$$T^{2022}|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}\begin{pmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{3}}{2} \\ -\frac{i}{2} \end{pmatrix} = \frac{\sqrt{3}}{2}|0\rangle - \frac{i}{2}|1\rangle$$

Es gibt noch eine andere Form dieser Aufgabenstellung, mit Tensorprodukten:

$$H \otimes H \otimes H \otimes H\left(|0\rangle \otimes |1\rangle \otimes |-\rangle \otimes \left(e^{-i\phi}|+\rangle\right)\right)$$

Hier kann man dann jedes Hadamard Gate auf je einen State anwenden:

$$H|0\rangle \otimes H|1\rangle \otimes H|-\rangle \otimes H\left(e^{-i\phi}|+\rangle\right)$$

$$|+\rangle \otimes |-\rangle \otimes H|1\rangle \otimes e^{-i\phi}|0\rangle$$

$$\frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2} \otimes |1\rangle \otimes e^{-i\phi}|0\rangle$$

$$\frac{|001\rangle - |011\rangle + |101\rangle - |111\rangle}{2} \otimes e^{-i\phi}|0\rangle$$

$$e^{-i\phi}\frac{|0010\rangle - |0110\rangle + |1010\rangle - |1110\rangle}{2}$$

Base Transform

Eine Basis ist eine Menge von Vektoren, die den Zustandsraum eines Quantensystems vollständig beschreiben. In der Quantenmechanik arbeiten wir häufig mit einem Hilbertraum, der ein abstrakter Vektorraum ist. Die Basisvektoren eines solchen Raums dienen als grundlegende Bausteine zur Beschreibung beliebiger Zustände.

Die gebräuchlichste Basis in der Quanteninformatik ist die Computational Basis. Für ein einzelnes Qubit sind die Basisvektoren:

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

Für ein System aus n Qubits ist die Computational Basis die Menge aller möglichen Kombinationen dieser Basisvektoren.

Basistransformationen sind Operationen, die einen Quantenzustand von einer Basis in eine andere Basis überführen. Dies ist nützlich, um Quantenzustände in unterschiedlichen Darstellungen zu analysieren oder zu manipulieren.

Eine alternative Basis zur Computational Basis ist die Hadamard-Basis, bestehend aus den Zuständen:

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

Die Transformation von der Computational Basis zur Hadamard-Basis erfolgt durch das Hadamard-Gate H: $H|0\rangle = |+\rangle, H|1\rangle = |-\rangle$.

Eine allgemeine Basistransformation wird durch eine unitäre Matrix U beschrieben, die die Basisvektoren der ursprünglichen Basis in die Basisvektoren der neuen Basis überführt. Wenn $\{|e_i\rangle\}$ die ursprüngliche Basis und $\{|f_i\rangle\}$ die neue Basis ist, dann gilt:

$$|f_i\rangle = U|e_i\rangle$$

Prove Entanglement

Wenn ein State nicht faktorisiert werden kann, ist er entangeled.

Wenn ein State nicht entagled ist, kann er geschrieben werden als: $|\psi\rangle = (a|0\rangle + b|1\rangle)\otimes(c|0\rangle + d|1\rangle)$, expandiert: $|\psi\rangle = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$. Dann die Faktoren des States überprüfen. Beispiel: $|\psi\rangle = \frac{1}{\sqrt{3}}|00\rangle + \frac{2}{3}|11\rangle$, wir benötigen die Koeffizienten $ad = 0, bc = 0, ac = \frac{1}{\sqrt{3}}, bd = \frac{2}{3}$, damit der State faktorisiert werden kann. Nun überprüfen wir die einzelnen Koeffizienten:

1. Wenn $a = 0$, dann $ac = 0$, was nicht $\frac{1}{\sqrt{3}}$ ist
2. Wenn $b = 0$, dann $bd = 0$, was nicht $\frac{2}{3}$ ist
3. Wenn $d = 0$, dann $bd = 0$, was nicht $\frac{2}{3}$ ist
4. Wenn $c = 0$, dann $ac = 0$, was nicht $\frac{1}{\sqrt{3}}$ ist

Es gibt also keine Koeffizienten Kombination, welche den State $|\psi\rangle$ beschreibt, daher ist dieser entangled.

Decomposition

Finden der Koeffizienten: $e_0 I + e_1 X + e_2 Y + e_3 Z$

Mithilfe der Gate Identities vereinfachen, dann erhält man im Idealfall am Ende ein einzelnes Gate oder Identity und dieser Koeffizient ist dann 1.

Steane-Code

Der Steane Code ist ein Quantum Error Correction Code. Er ist ein bestimmter Stabilisierer-Typ, welcher jede Anzahl von 1-Qubit-Fehlern korrigieren kann (Bit und Phase Flip). Der Code enkodiert ein logisches Qubit in 7 physische Qubits.

Wenn ein Fehler auftritt, verändert der Steane Code den Status der Qubits und durch das Messen der Stabilisierer, kann man den Typ und den Ort des Fehlers feststellen. Der Fehler kann korrigiert werden, in dem man den passenden Pauli Operator anwendet.

Zeigen, das ein Gate transversely auf dem Steane Code implementiert werden kann.

Z-Gate:

$Z|0\rangle_L = |0\rangle_L$, wobei $|0\rangle_L$: #$|1\rangle$ gerade, daraus folgt, - kürzt sich weg

$Z|1\rangle_L = -|1\rangle_L$, wobei $|1\rangle_L$: #$|1\rangle$ ungerade, daraus folgt, - bleibt

H-Gate: Siehe Lösung Übung 10 Aufgabe 3 a) / Testklausur SoSe 23 Lösung

CNOT-Gate: Siehe Lösung Übung 10 Aufgabe 3 b) / Testklausur SoSe 23 Lösung

T-Gate: Siehe Lösung Übung 10 Aufgabe 3 c) / Testklausur SoSe 23 Lösung

Y-Gate: Wir wenden das Y-Gate auf jedes Qubit an, die Phasen lösen sich auf aber die Invertierung bleibt

i^1	i^2	i^3	i^4	i^5	i^6	i^7	i^8
i	-1	$-i$	1	i	-1	$-i$	1

$$Y^{\otimes 7}|0\rangle_L = \frac{1}{\sqrt{8}}((-i)|1111111\rangle + (-i)|0101010\rangle + (-i)|1001100\rangle + (-i)|0011001\rangle + (-i)|1110000\rangle + (-i)|0000111\rangle + (-i)|1100011\rangle + (-i)|1010101\rangle) = -i|1\rangle_L$$

$$Y^{\otimes 7}|1\rangle_L = \frac{1}{\sqrt{8}}(i|0000000\rangle + i|1010101\rangle + i|0110011\rangle + i|1100110\rangle + i|0001111\rangle + i|1111000\rangle + i|0011100\rangle + i|0101010\rangle) = i|0\rangle_L$$

The logical Y_L is then $-Y^{\otimes 7}$.

X-Gate: Wir wenden das X Gate auf jedes Qubit an:

$$X^{\otimes 7}|0\rangle_L = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0000111\rangle + |1100011\rangle + |1010101\rangle) = |1\rangle_L$$

$$X^{\otimes 7}|1\rangle_L = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1111000\rangle + |0011100\rangle + |0101010\rangle) = |0\rangle_L$$

S-Gate: ähnliche Argumentation wie Y, wenn man S auf ein code element applied bekommt man für 0L immer $i^4 = i^0 = 1$ und daher S_L auf $0_L = 0_L$. bei 1_L bekommt man $i^3 = i^7 = -i$ und daher $-i1_L$. Insgesamt bekommt man wie bei Y ein $-S_L$ Gate.

Rotation-Gates: identisch zum T Gate

Shor's Code

Der 9-Qubit Shor-Code, ist ein Quantenfehlerkorrekturcode, der vor Bit-Flip- und Phase-Flip-Fehlern schützt. Er kombiniert Wiederholungscodes und nutzt sowohl die klassische als auch die quantenmechanische Fehlerkorrektur.

Zunächst wird ein einzelnes Qubit 3 mal wiederholt, und dieses dreifache Qubit wird nochmals 3 fach wiederholt, so erhalten wir:

$$|0\rangle_L = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

$$|1\rangle_L = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

Bit-Flip-Fehlerkorrektur: Jeder der drei Blöcke von drei Qubits wird auf Bit-Flip-Fehler geprüft und korrigiert.

Phasenfehlerkorrektur: Die gesamte Blockstruktur wird auf Phasenfehler geprüft und korrigiert.

Swap Gate

Siehe Testklausur SoSe 2023 Lösung, Lösung mithilfe von Logik-Tabellen

Swap Entanglement

Siehe Altklausur SoSe 2022 Lösung.

Prove Unitarity

Um zu zeigen, dass eine Matrix mit globaler Phase Unitary ist, $M = e^{i\theta}U$, reicht es zu zeigen, dass U unitary ist.

1. U ist Unitary, wenn $U^\dagger U = I$
2. Wir berechnen die hermetische Konjugation: $M^\dagger = \left(e^{i\theta}U\right)^\dagger = e^{-i\theta}U^\dagger$
3. So können wir die Unitary überprüfen: $M^\dagger M = \left(e^{i\theta}U\right)^\dagger\left(e^{i\theta}U\right) = e^{-i\theta}e^{i\theta}U^\dagger U = U^\dagger U = I$

Um U zu überprüfen, berechnen wir die resultierende Matrix in der computational Basis.

Beispiel:

$$|+\rangle\langle+| + |-\rangle\langle-| = \frac{1}{2}(|0\rangle\langle0| + |0\rangle\langle1| + |1\rangle\langle0| + |1\rangle\langle1|) + \frac{1}{2}(|0\rangle\langle0| - |0\rangle\langle1| - |1\rangle\langle0| + |1\rangle\langle1|)$$
$$= \frac{1}{2}(2|0\rangle\langle0| + 2|1\rangle\langle1|) = |0\rangle\langle0| + |1\rangle\langle1| = I$$

Quantenorakel

Ein Orakel U_f für eine Funktion $f: \{0,1\}^n \to \{0,1\}$ wird normalerweise als unitäre Operation dargestellt, die auf einem n-Qubit-Eingangszustand x und einem einzelnen Qubit für das

Ergebnis y angewendet wird. Der Orakel-Operator U_f ist so definiert, dass er den Zustand $|x, y\rangle$ in $|x, y \oplus f(x)\rangle$ transformiert, wobei $\oplus$ die Addition modulo 2 ist (XOR-Operation).

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$$

Quantenparallelismus

Quantenparallelismus entsteht durch die Superposition von Qubits und die parallele Verarbeitung dieser Zustände.

Wenn man ein System aus mehreren Qubits hat, kann jeder Qubit in einer Superposition sein. Für ein System aus n Qubits kann der Gesamtzustand als Superposition von 2^n Zuständen geschrieben werden:

$$|\psi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle$$

Wobei $|x\rangle$ die Basiszustände des Systems sind. (Beispiel $n = 2$: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$).

Eine Quantenoperation (wie ein Quantenorakel) kann auf einem Superpositionszustand angewendet werden. Dies bedeutet, dass die Operation gleichzeitig auf alle Basiszustände in der Superposition angewendet wird. Zum Beispiel, wenn ein Orakel U_f auf den Superpositionszustand angewendet wird:

$$U_f\left(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle\right) = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1} U_f|x\rangle$$

Hier wird U_f parallel auf alle Basiszustände $|x\rangle$ angewendet.

Shor‘s Algorithmus

Der Shor-Algorithmus, ist ein quantenmechanischer Algorithmus zur Faktorisierung großer Zahlen. Er ist exponentiell schneller als die besten bekannten klassischen Algorithmen und somit die Grundlage für die Sicherheit vieler kryptographischer Systeme, er bricht RSA.

1. Problemformulierung: Eine Zahl N in zwei nicht-triviale Faktoren zerlegen
2. Zufällige Auswahl: Wähle eine zufällige Zahl a mit $1 < a < N$, die teilerfremd zu N ist (also $gcd(a, N) = 1$)
3. Periodenbestimmung: Finde die kleinste Periode r der Funktion $f(x) = a^x \; mod \; N$, sodass $a^r \equiv 1 \; mod \; N$. Hierfür wird die Quatum Fourier Transformation verwendet.
4. Faktoren berechnen: Wenn r eine gerade Zahl ist, berechne die Kandidaten für die Faktoren von N als gcd $(a^{\frac{r}{2}} \pm 1, N)$

Quantum Fourier Transform

Die Quantum Fourier Transformation ist eine spezielle Quantenoperation, die Eingabequbits in eine neue Basis transformiert, ähnlich wie eine Drehung in einen neuen Koordinatenraum. Diese Transformation hat die Eigenschaft, dass sie periodische Muster in den Eingabedaten aufdeckt.

Die QFT arbeitet auf einem Register von n Qubits. Angenommen, der Zustand des Registers ist $|x\rangle$, wobei x eine Binärzahl zwischen 0 und $2^n - 1$ ist. Die QFT transformiert diesen Zustand $|x\rangle$ in einen neuen Zustand $|y\rangle$ durch die Anwendung einer bestimmten Quantenoperation.

1. Startzustand $|x\rangle$, mit x eine Binärzahl
2. Die QFT wendet Hadamard-Gates und kontrollierte Rotationen an, um den Zustand $|x\rangle$ in einen neuen Zustand zu transformieren
3. Der resultierende Zustand ist eine Superposition aller möglichen Zustände $|y\rangle$, wobei jede Amplitude durch eine bestimmte Phase gewichtet ist.

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

Wobei $N = 2^n$ die Anzahl der möglichen Zustände ist, x der Eingabewert in Binärdarstellung und k die resultierenden Zustände.

Grover

Der Grover-Algorithmus, ist ein quantenmechanischer Suchalgorithmus, der eine unstrukturierte Datenbank mit N Einträgen in $\mathcal{O}(\sqrt{N})$ Schritten durchsuchen kann. Er verbessert die klassische Suchzeit, die $\mathcal{O}(N)$ beträgt, erheblich.

1. Beginne mit einem gleichmäßigen Superpositionszustand aller möglichen Einträge.
2. Grover-Iteration: Folgende Schritte etwa $\sqrt{N}$ mal wiederholen
    a. Markieren: Orakeloperation anwenden, die den gesuchten Eintrag markiert durch das Umkehren der Phase
    b. Diffusion: Verstärke die Amplitude des markierten Zustands durch eine Diffusionsoperation
3. Messung durchführen, um den gesuchten Eintrag zu finden.

Initial Superposition $|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$

Oracle Application $U_f|x\rangle = \begin{cases} -|x\rangle, if\ x = x_t \\ |x\rangle, if\ x \neq x_t \end{cases}$

Diffusion Operator $U_R = 2|\psi_0\rangle\langle\psi_0| - I$

Grover Operator $G = U_R \cdot U_f$

State after k Iterations $|\psi_k\rangle = G^k|\psi_0\rangle$

Deutsch's Algorithm

Deutsch's Algorithmus überprüft eine Funktion f: $\{0,\ 1\} \rightarrow \{0,\ 1\}$ und entscheidet, ob diese konstant oder ausgewogen ist.

1. Startzustand $|0\rangle|1\rangle$, darauf Hadamard Gate anwenden
2. Ein Orakel U_f implementiert die Funktion f und wirkt auf den Zustand, indem es das zweiten Qubit, bedingt durch $f(x)$ flippt.
3. Hadamard auf das erste Qubit anwenden, Qubit messen, wenn das Ergebnis 0 ist, ist die Funktion konstant, bei 1 ausgewogen.

Deutsch-Josza Algorithm

Der Deutsch-Josza Algorithmus ist eine Verallgemeinerung von Deutschs Algorithmus für Funktionen mit mehreren Eingangsbits und zeigt eine exponentielle Beschleunigung gegenüber klassischen Algorithmen.

Wir überprüfen eine Funktion $f\colon \{0,\ 1\}^n \ \rightarrow \ \{0,\ 1\}$ und entscheidet, ob diese konstant oder ausgewogen ist.

1. Startzustand $|0\rangle^{\otimes n}|1\rangle$, darauf Hadamard anwenden
2. Ein Orakel U_f implementiert die Funktion f und wirkt auf den Zustand, indem es das letzte Qubit, bedingt durch $f(x)$ flippt.
3. Dann Hadamard auf die ersten n Bits anwenden und diese Messen. Wenn das Ergebnis $|0\rangle^{\otimes n}$ ist, ist die Funktion konstant, andernfalls ausgewogen

Bernstein-Vazirani Algorithm

Der Bernstein-Vazirani Algorithmus findet eine geheime Zeichenkette a in einer linearen Funktion $f(x) = a \cdot x \ mod\ 2$ mit einer einzigen Abfrage des Orakels.

Eine Funktion $f\colon \{0,1\}^n \rightarrow \{0,1\}$ ist definiert als $f(x) = a \cdot x \ mod\ 2$ für eine unbekannte Zeichenkette a.

1. Startzustand $|0\rangle^{\otimes n}|1\rangle$, darauf Hadamard anwenden
2. Ein Orakel U_f implementiert die Funktion f und wirkt auf den Zustand.
3. Dann Hadamard auf die ersten n Bits anwenden und diese Messen, die ersten n Qubits liefern direkt die Zeichenkette a.

Simon's Algorithm

Simons Algorithmus findet eine geheime Zeichenkette s in einer periodischen Funktion $f\colon \{0,1\}^n \rightarrow \{0,1\}^m$, die $f(x) = f(x \oplus s)$ erfüllt.

1. Startzustand $|0\rangle^{\otimes n}|1\rangle^{\otimes m}$, Hadamard auf die ersten n Qubits anwenden
2. Ein Orakel U_f implementiert die Funktion f und wirkt auf den Zustand.
3. Dann Hadamard auf die ersten n Bits anwenden und diese Messen. Dadurch erhählt man ein lineares Gleichungssystem. Löst man dieses, erhält man s.

Quantum Key Distribution

Quantum Key Distribution (QKD) ist eine Methode zur sicheren Übertragung von Verschlüsselungsschlüsseln unter Verwendung der Prinzipien der Quantenmechanik. Es ermöglicht zwei Parteien, einen gemeinsamen geheimen Schlüssel zu generieren, der anschließend für die symmetrische Verschlüsselung von Nachrichten verwendet werden kann. Der wichtigste Aspekt von QKD ist, dass es die Sicherheit des Schlüsselaustauschs garantiert, selbst gegenüber einem mächtigen Angreifer mit unbegrenzten Rechenressourcen.

Bell States

Bell-Zustände sind spezielle Quantenverschränkungszustände von zwei Qubits, die eine maximales Entanglement aufweisen.

Bell-Zustände (Bell-Basis):

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

Bell-Zustände sind maximal verschränkte Zustände. Dies bedeutet, dass die Messung eines Qubits den Zustand des anderen Qubits sofort bestimmt, unabhängig von der Entfernung zwischen den Qubits.

Außerdem sind Sie orthonormal zu einander und werden erhalten durch Anwendung von Hadamard und CNOT.

Linearity

Linearität ist ein zentrales Prinzip in der Quanteninformatik, das sicherstellt, dass Quantenoperationen auf Superpositionen angewendet werden können und die parallele Verarbeitung von Zuständen ermöglicht wird.

In der Quantenmechanik müssen Operationen unitär sein, was bedeutet, dass sie linear und normerhaltend sind. Dies stellt sicher, dass die Gesamtwahrscheinlichkeit erhalten bleibt, was physikalisch bedeutet, dass die Summe der Wahrscheinlichkeiten aller möglichen Messresultate immer 1 bleibt.

Superdense Coding

Superdense Coding nutzt die Verschränkung von Qubits, um die Informationskapazität zu erhöhen. Zwei Parteien teilen ein verschränktes Qubit-Paar. Durch geeignete Manipulation dieses verschränkten Paares kann Alice zwei klassische Bits Information an Bob senden, indem sie nur ein Qubit überträgt.

1. Alice und Bob teilen ein verschränktes Qubit-Paar. Ein typischer verschränkter Zustand ist der Bell-Zustand $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
2. Alice wählt die zu sendenden zwei klassischen Bits (00, 01, 10 oder 11) aus. Abhängig von den gewählten Bits führt Alice eine von vier möglichen Quantenoperationen auf ihr Qubit aus: 00 (I), 01 (X), 10 (Z), 11 (Y)
3. Alice sendet ihr Qubit an Bob über einen Quantenkanal.
4. Bob hat nun beide Qubits des verschränkten Paares. Bob wendet zunächst ein CNOT-Gate an, wobei sein Qubit das Kontrollqubit und das von Alice gesendete Qubit das Zielqubit ist. Danach wendet Bob ein Hadamard-Gate auf sein eigenes Qubit an. Bob misst beide Qubits in der Computational Basis (Standardbasis) und erhält die ursprünglichen zwei Bits, die Alice senden wollte.