

Quantencomputing

Modul 7271

Martin Rehberg

DB Systel GmbH / Hochschule RheinMain

Inhaltsverzeichnis

1 Grundlagen

- Einleitung & Ein-Qubit Systeme
- n -Qubit Systeme & Grundlegende Schaltkreise
- Verschränkung, Deutsch-Jozsa, Bernstein-Vazirani & Simon

2 Quantensuche

- Algorithmus von Grover
- Varianten der Quantensuche

3 Fouriertransformation & Shor

- RSA & periodische Funktionen
- Diskrete- & Quanten-Fouriertransformation
- Algorithmus von Shor

4 Ausblick: QEC & QKD

Inhaltsverzeichnis

1 Grundlagen

- Einleitung & Ein-Qubit Systeme
- n -Qubit Systeme & Grundlegende Schaltkreise
- Verschränkung, Deutsch-Jozsa, Bernstein-Vazirani & Simon

2 Quantensuche

- Algorithmus von Grover
- Varianten der Quantensuche

3 Fouriertransformation & Shor

- RSA & periodische Funktionen
- Diskrete- & Quanten-Fouriertransformation
- Algorithmus von Shor

4 Ausblick: QEC & QKD

Einleitung

Themengebiet Quantencomputing:

- Quantenalgorithmen entwerfen
- Quantenalgorithmen untersuchen (insb. Laufzeit)
- Quantenkomplexitätstheorie

Einleitung

Themengebiet Quantencomputing:

- Quantenalgorithmen entwerfen
- Quantenalgorithmen untersuchen (insb. Laufzeit)
- Quantenkomplexitätstheorie

Ziel der Vorlesung:

- grundlegende Algorithmen auf Quantencomputern verstehen
 - physikalischen Grundlagen als *gegeben* annehmen
 - Mathematik werden wir nach Bedarf erarbeiten
- Anwendungen, insb. mit Blick auf Verschlüsselungsverfahren
- Ausblick: Quantenfehlerkorrektur, Quantum Key Distribution

Einleitung

Literatur:¹

- Matthias Hanneister - Quantencomputing verstehen (**Hauptquelle**), 5. Auflage, Springer, 2018.
- Artuhr Pittenger - An Introduction to Quantum Computing Algorithms, Birkhäuser, 2001.
- Michael Nielsen, Isaac Chuang - Quantum Computation and Quantum Information, 10. Auflage, Cambridge University Press, 2010.
- Wolfgang Scherer - Mathematics of Quantum Computing, 2. Auflage, Springer, 2019.

¹ verwendete Grafiken sind allesamt dem Buch von M. Hohmeister oder Wikipedia (public domain) entnommen

Einleitung

Klassische Welt

- mechanische Rechenmaschinen
 - Difference Engine (Auswertung von Polynomen), Analytical Engine (Gleichungssysteme lösen) - Charles Babbage
 - Schachmaschine - Leonardo Quevedo
- elektromechanische Rechenmaschinen
 - Z3 (Gleitkommaarithmetik), Z4 - Konrad Zuse
 - Kryptoanalyse (Lorenz-Schlüsselmaschine) - Colossus
- *moderne* Rechenmaschinen

Einleitung

Beobachtung

Ein **klassisches Bits** kann genau zwei unterschiedliche Zustände annehmen: 0 und 1. Sie haben zwei wesentliche Eigenschaften

- **Realismus:** Der Wert eines Bits ist zu jedem Zeitpunkt der Berechnung eindeutig bestimmt, d.h. entweder 0 oder 1. Er kann ausgelesen werden und der Prozess des Auslesens ändert den Wert des Bits nicht.
- **Lokalität:** Wird der Wert eines bestimmten einzelnen Bits verändert, so ändert das nicht den Wert *irgendeines* anderen Bits.

Einleitung

Quantenwelt

- Quantencomputer rechnen mit Quantenbits
- Quantenbits folgen den Gesetzen der Quantenmechanik (und können bspw. nicht *kopiert* werden)
- Quantenbits sind in einem Zustand der *Superposition*, d.h. sind von der Form $\alpha|0\rangle + \beta|1\rangle$
- Quantenbits können in einem *verschränkten* Zustand sein

Einleitung

Beobachtung

Ein **Quantenbit** ist in einem Zustand der Superposition. Im Vergleich zum klassischen Bit stellen wir fest:

- **Veränderung beim Messen:** Wird ein Quantenbit gemessen, so wird der Zustand der Superposition aufgehoben und das Quantenbit wechselt in einen der beiden (klassischen) Zustände 0 oder 1. Durch den Messvorgang wird das Quantenbit mit dem entsprechenden Werte 0 oder 1 überschrieben.
- **Verschränkung:** Die Veränderung eines Quantenbits kann unmittelbar (also im selben Augenblick) die Eigenschaft eines anderen Quantenbits verändern.

Einleitung

Quantencomputing hat bereits weitreichende Folgen

- Kryptographie (RSA & ECDH)
- Optimierungsverfahren & ML (Logistik & Transport)
- Chemie- & Pharmaindustrie (Material & Medikamente)

Einleitung

Quantencomputing hat bereits weitreichende Folgen

- Kryptographie (RSA & ECDH)
- Optimierungsverfahren & ML (Logistik & Transport)
- Chemie- & Pharmaindustrie (Material & Medikamente)

Es gibt aber nicht nur Vorteile:

- No-Cloning Theorem
- Quantencomputer können **NP-vollständige Probleme** (vermutlich) nicht effizient lösen
- Fehlerkorrektur

Einleitung - Komplexität

Rückblick & Ausblick Schwierigkeit von Berechnungsprobleme

Beispiel: Zur Addition zweier n -Bit Zahlen mittel *Carry-Ripple Addierer* wird ein Halbaddierer (ein XOR, ein AND) und $n - 1$ Volladdierer (zwei XOR, zwei AND, ein OR) benötigt; insgesamt $5n - 3$ logische Gatter. Wächst die Länge der binären Zahlen, dann auch die Anzahl der logischen Gatter *linear* in n .

Landau Notation

Sei $f, g : \mathbb{N} \rightarrow \mathbb{N}$. Wir schreiben $f(n) = O(g(n))$, wenn eine Konstante $c > 0$ und ein $n_0 \in \mathbb{N}$ existiert, sodass $f(n) \leq c \cdot g(n)$ für alle $n \geq n_0$ gilt.

Hier, $5n - 3 = O(n^2)$ für alle $n \geq 5$.

Einleitung - Komplexität

Die Anwendung eines logischen Gatter benötigt eine gewisse Zeit.
Von einem Algorithmus der $O(n)$ logische Gatter benötigt, sagen wir er läuft in **Zeit** $O(n)$.

Ein Algorithmus ist **effizient**, wenn seine Laufzeit (in Abhängigkeit der Inputgröße) durch ein Polynom beschrieben werden kann. Die Anzahl der Gatter skaliert **polynomial** in n , etwa $5n - 3$, n^2 , \sqrt{n} oder n^{100} .

Ein Algorithmus ist **ineffizient**, wenn er mehr als polynomielle Laufzeit benötigt, etwa

- **exponentielle Laufzeit**, wie 2^n oder $\exp\left(\frac{n}{1000}\right)$
- **superpolynomielle Laufzeit**, d.h. weniger als exponentielle Zeit aber mehr als polynomielle Zeit, wie $2^{n^{1/3}}$

Einleitung - Komplexität

Probleme die von einem klassischen Computer effizient (d.h. in Polynomialzeit) gelöst werden können, ordnen wir der Komplexitätsklasse **P** zu.

Beispiel:

- Addition zweier Zahlen
- Bestimmung ob eine Zahl eine Primzahl ist

Einleitung - Komplexität

Probleme für die eine gegebene Lösung effizient überprüft werden kann, werden der Komplexitätsklasse **NP** zugeordnet.

Beispiel:

- Faktorisierung einer Zahl
- Überprüfung ob zwei gegebene Graphen äquivalent sind

Einleitung - Komplexität

Probleme für die eine gegebene Lösung effizient überprüft werden kann, werden der Komplexitätsklasse **NP** zugeordnet.

Beispiel:

- Faktorisierung einer Zahl
- Überprüfung ob zwei gegebene Graphen äquivalent sind

Unter den Problemen in **NP** gibt es besondere Probleme, die wir **NP-vollständig** nennen. Können wir ein **NP-vollständiges** Problem effizient lösen, dann können wir **jedes** Problem in **NP** effizient lösen. Dazu gehören

- TSP (Traveling Salesperson Problem)
- spieltheoretische Probleme wie $n \times n$ Sudoku

Einleitung - Komplexität

Eine speicherbezogene Klasse ist **PSPACE**, die alle Probleme beinhaltet die auf einem klassischen Computer mit polynomial viel Speicher gelöst werden können, wobei an die Zeit keine Schranken vorgegeben werden.

Beispiel: Gewinnstrategie für Zwei-Personen Spiele wie Dame auf einem $n \times n$ Spielbrett.

Es gilt

- $\mathbf{P} \subset \mathbf{NP}$
- Umkehrung: \mathbf{P} vs. \mathbf{NP} Problem
- $\mathbf{NP} \subset \mathbf{PSPACE}$
- Vermutung: $\mathbf{P} \neq \mathbf{NP}$, $\mathbf{P} \neq \mathbf{PSPACE}$, $\mathbf{NP} \neq \mathbf{PSPACE}$

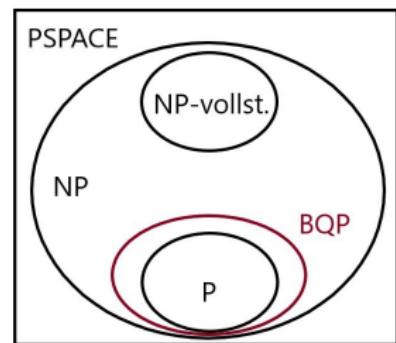
Einleitung - Komplexität

Die (begründete) Hoffnung ist das Quantencomputer Probleme effizient lösen können, die auf einem klassischen Computer nicht effizient gelöst werden können.

Dafür spricht

- Algorithmus von Shor
- Random Circuit Sampling

Ein Quantencomputer liefert in beiden Fällen superpolynomielle Beschleunigung.

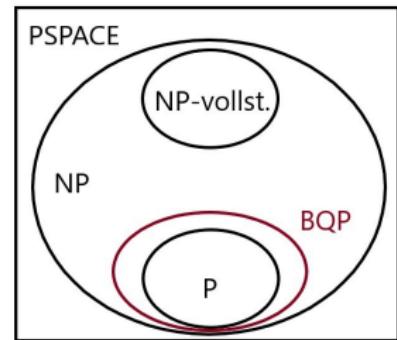


Einleitung - Komplexität

Für andere Probleme liefert ein Quantencomputer eine (beweisbar) polynomielle Beschleunigung (ggf. schon anwendungsrelevant).

Probleme die von einem Quantencomputer effizient gelöst werden können, liegen in **BQP**.
Quantencomputer können klassische Computer effizient simulieren, d.h. $P \subset BQP$.

Welche Probleme liegen in **BQP**?
Wie viel „größer“ als **P** ist **BQP**?

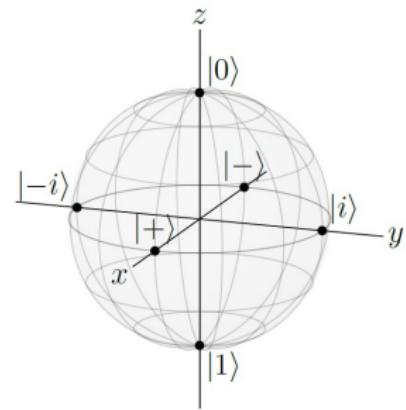


Ein-Qubit Systeme

Ein Qubit kann zwei Grundzustände annehmen: $|0\rangle$ und $|1\rangle$.

Wir identifizieren diese mit dem Nordpol $(1, 0, 0)$ und Südpol $(0, 0, -1)$ der *Blochschen Sphäre*.

Zustände eines Qubits können auch Superpositionen von $|0\rangle$ und $|1\rangle$ sein.

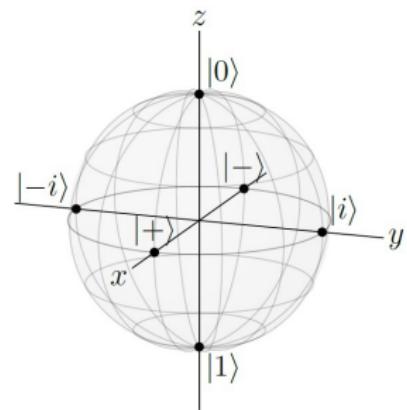


Ein-Qubit Systeme

Einige auf Äquatorebene gegenüberliegende Zustände sind von besonderer Bedeutung:

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

Grundsätzlich kann ein Qubit jeder Punkt auf der Blochschen Sphäre sein.



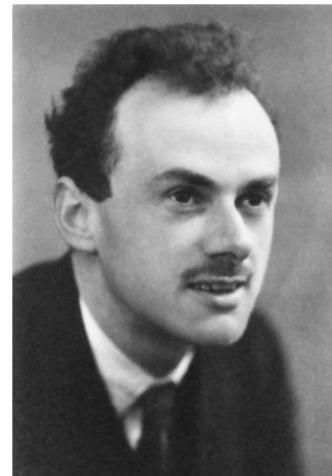
Ein-Qubit Systeme

Für die Beschreibung quantenmechanischer Zustände verwendet man die auf Paul Dirac zurückgehende *ket-Notation*.

Definition (Quantenbit)

Ein Quantenbit (*Qubit*) nimmt Zustände der Form $\alpha|0\rangle + \beta|1\rangle$ mit $\alpha, \beta \in \mathbb{C}$ an. Die Zahlen α, β heißen *Amplituden* und genügen der Bedingung $|\alpha|^2 + |\beta|^2 = 1$.

Klassische Bits: $|0\rangle, |1\rangle$.



Paul Dirac

Ein-Qubit Systeme

Übung:

1. Überprüfen Sie, ob es sich bei $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ und $\frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$ um zulässige Zustände handelt.

Ein-Qubit Systeme

Übung:

1. Überprüfen Sie, ob es sich bei $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ und $\frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$ um zulässige Zustände handelt.
2. Gilt für ein Qubit $|\alpha|^2 + |\beta|^2 = 1$, nennt man es *normalisiert*. Bestimmen Sie die *Normalisierungskonstante* A , sodass $A(\sqrt{2}|0\rangle + i|1\rangle)$ normalisiert ist.

Ein-Qubit Systeme

Übung:

1. Überprüfen Sie, ob es sich bei $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ und $\frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$ um zulässige Zustände handelt.
2. Gilt für ein Qubit $|\alpha|^2 + |\beta|^2 = 1$, nennt man es *normalisiert*. Bestimmen Sie die *Normalisierungskonstante* A , sodass $A(\sqrt{2}|0\rangle + i|1\rangle)$ normalisiert ist.

Während wir den Zustand klassischer Bits durch *lesen* feststellen können, ist das bei Qubits nicht ohne Weiteres möglich. Bei Qubits müssen wir *messen* und das Messergebnis hängt von den Amplituden ab.

Ein-Qubit Systeme

Messen eines Quantenbits

Messen wir ein Qubit im Zustand $\alpha|0\rangle + \beta|1\rangle$, wird die Superposition zerstört. Anschließend ist es mit Wahrscheinlichkeit $|\alpha|^2$ im Zustand $|0\rangle$ und mit Wahrscheinlichkeit $|\beta|^2$ im Zustand $|1\rangle$. Diesen Zustand nach dem Messen können wir beobachten.

Beispiel: Das Qubit $\frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$ ist nach dem Messen mit Wahrscheinlichkeit $1/3$ im Zustand $|0\rangle$ und mit Wahrscheinlichkeit $2/3$ im Zustand $|1\rangle$.

Ein-Qubit Systeme

Übung:

1. Was beobachten Sie beim Messen der Qubits $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ und $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$?
2. Bestimmen Sie die Messwahrscheinlichkeiten des Qubits $\frac{2}{3}|0\rangle + \frac{1-2i}{3}|1\rangle$.

Ein-Qubit Systeme

Übung:

- Was beobachten Sie beim Messen der Qubits $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ und $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$?
- Bestimmen Sie die Messwahrscheinlichkeiten des Qubits $\frac{2}{3}|0\rangle + \frac{1-2i}{3}|1\rangle$.

Wir unterscheiden verschiedene Messbasen (entsprechend der Lage der x, y, z -Achse auf der Blochschen Sphäre):

- Z-Basis (bzw. *computational basis*) $\{|0\rangle, |1\rangle\}$
- X-Basis (bzw. *Hadamard-Basis*) $\{|+\rangle, |-\rangle\}$
- Y-Basis (bzw. *LR-Basis*) $\{|i\rangle, |-i\rangle\}$

Messwahrscheinlichkeiten sind die jeweiligen Amplituden bestimmt.

Ein-Qubit Systeme

Wir bauen unsere vorheriges Beispiel weiter aus und vergleichen für bel. $\theta \in \mathbb{R}$

$$\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle \quad \text{und} \quad e^{i\theta} \left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle \right)$$

bzgl. einer Messung in Z-Basis. Hier ergibt sich für das zweite Qubit $|0\rangle$ bzw. $|1\rangle$ mit einer Wahrscheinlichkeit von

$$\left| e^{i\theta} \cdot \frac{\sqrt{3}}{2} \right|^2 = \frac{3}{4} \quad \text{bzw.} \quad \left| e^{i\theta} \cdot \frac{1}{2} \right|^2 = \frac{1}{4}$$

Ein-Qubit Systeme

denn $|\exp(i\theta)|^2 = 1$ für alle $\theta \in \mathbb{R}$. Die Messwahrscheinlichkeiten unterscheiden sich also nicht von denen für das erste Qubit.

Wir nennen $\exp(i\theta)$ eine *globale Phase*. Beide Qubits repräsentieren denselben Punkt auf der Blochschen Sphäre. Globalen Phasen sind vernachlässigbar und wir identifizieren Qubits die sich nur um eine globale Phase unterscheiden miteinander, und schreiben

$$e^{i\theta} \left(\frac{\sqrt{3}}{2} |0\rangle + \frac{1}{2} |1\rangle \right) \equiv \frac{\sqrt{3}}{2} |0\rangle + \frac{1}{2} |1\rangle.$$

Ein-Qubit Systeme

Globale Phasen sollten nicht mit *relative Phasen* verwechselt werden. Beim Vergleich von

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

$$|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{2}}|1\rangle)$$

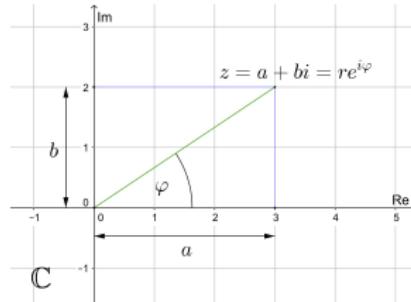
ist $\exp(i\frac{\pi}{2})$ eine relative Phase und die Qubits $|+\rangle, |i\rangle$ gehören zu verschiedenen Punkten auf der Blochschen Sphäre.

Die Darstellung von Qubits auf einer Kugel (*sphärische Koordinaten*) ist eine Erweiterung der aus Analysis bekannten Darstellung mittel Polarkoordinaten.

Ein-Qubit Systeme

Erinnerung: Jede komplexe Zahl $z \in \mathbb{C}$ kann in der Form $z = a + ib$ mit $a, b \in \mathbb{R}$ und $i := \sqrt{-1}$ geschrieben werden. Die Zahl $\bar{z} := a - ib$ heißt die *Konjugierte* von z . Der *Betrag* einer komplexen Zahl ist $|z| := \sqrt{a^2 + b^2}$.

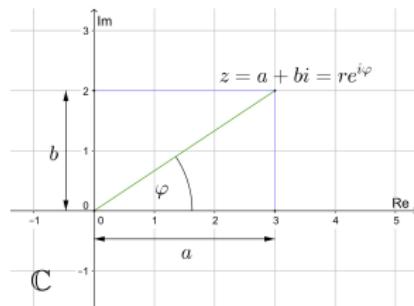
Die *Polarkoordinatendarstellung* einer komplexen Zahl ist $z = re^{i\varphi}$, wobei r der Betrag ist und φ die *Phase*.



Ein-Qubit Systeme

Wissen: Gilt $|z| = |z'|$ für $z \neq z'$ mit $z, z' \in \mathbb{C}$, so unterscheiden sich die komplexen Zahlen nur in der Phase.

Wie selbstverständlich identifizieren wir \mathbb{C} mit \mathbb{R}^2 mittels $1 = (1, 0)$ und $i = (0, 1)$.



Ein-Qubit Systeme

Identifizieren wir $|0\rangle$ mit $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und $|1\rangle$ mit $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, können wir ein Qubit als Kombination linear unabhängiger Vektoren darstellen:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha|0\rangle + \beta|1\rangle.$$

Unter der Bedingung $|\alpha|^2 + |\beta|^2 = 1$ an die Amplituden $\alpha, \beta \in \mathbb{C}$ erhalten wir, dass ein Qubit ein Vektor aus \mathbb{C}^2 der Länge 1 ist.

D.h. die Superposition ist eine *Linearkombination* der klassischen (nicht überlagerten) Zustände $|0\rangle$ und $|1\rangle$.

Achtung: $\alpha, \beta \in \mathbb{C}$, d.h. wie befinden uns im \mathbb{C}^2 .

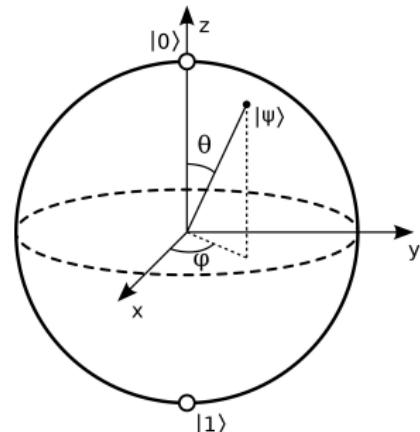
Ein-Qubit Systeme

Mittels

$$\alpha = \cos \frac{\theta}{2}, \quad \beta = e^{i\varphi} \sin \frac{\theta}{2}$$

können wir uns das Qubit
 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ auf der
Blochschen Sphäre veranschaulichen.

Rechenoperationen auf einem Qubit
sind Rotationen auf der Blochschen
Sphäre.



Ein-Qubit Systeme

Wir wollen verstehen wie man

- Rechenoperationen auf einem Ein-Qubit System ausführt
- Register von mehreren Qubits generiert
- mit *Quantengattern* Operationen auf mehreren Qubits ausführt
- *Quantenalgorithmen* (Schaltkreise) entwickeln
- Register von Qubits misst

und benötigen dafür einen methodischen *Werkzeugkasten* (hier:
Lineare Algebra).

Lineare Algebra

Wir wiederholen einige Grundlagen aus der Linearen Algebra und machen uns mit der Notation im Kontext von Qubits vertraut.

Die Schreibweise $|0\rangle$ und $|1\rangle$ für Qubits ist eine abkürzende Notation für Spaltenvektoren (hier: Basisvektoren), d.h.

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

Ein (generisches) Qubit $|\psi\rangle$ mit Amplituden α, β ist demnach ein Vektor

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

Lineare Algebra

Beispiel:

$$|i\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{i}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix}$$

Die Transponierte eines Spaltenvektors ist ein Zeilenvektor

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}^T = (\alpha \ \beta)$$

und die Konjugierte des transponierten Vektors ist

$$\left(\begin{pmatrix} \alpha \\ \beta \end{pmatrix}^T \right)^\dagger = (\bar{\alpha} \ \bar{\beta}).$$

Dagger
Transponieren
+
Konjugieren
=>Dagger-Notation

Lineare Algebra

// Komplexe Zahl $z = a + ib$
 $\bar{z} = a - ib$

Letzteres wird so häufig verwendet, dass es im Rahmen der bra-ket Notation eine eigene Symbolik hat

$$\langle \psi | = (\overline{\alpha} \quad \overline{\beta}) .$$

Beispiel:

$$| i \rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{pmatrix} \quad \text{und} \quad \langle i | = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-i}{\sqrt{2}} \end{pmatrix}$$

Lineare Algebra

Für $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ ist $\langle 0| = (1 \ 0)$ und $\langle 1| = (0 \ 1)$, d.h.

$$\langle\psi| = (\bar{\alpha} \ \bar{\beta}) = \bar{\alpha}(1 \ 0) + \bar{\beta}(0 \ 1) = \bar{\alpha}\langle 0| + \bar{\beta}\langle 1|.$$

Beispiel:

$$|i\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle \quad \text{dann} \quad \langle i| = \frac{1}{\sqrt{2}}\langle 0| - \frac{i}{\sqrt{2}}\langle 1|$$

Lineare Algebra

Wir fassen unsere Beobachtungen zusammen:

bra-ket Notation

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \iff \langle\psi| = (\bar{\alpha} \quad \bar{\beta})$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \iff \langle\psi| = \bar{\alpha}\langle 0| + \bar{\beta}\langle 1|$$

Lineare Algebra

Wir fassen unsere Beobachtungen zusammen:

bra-ket Notation

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \iff \langle\psi| = (\bar{\alpha} \quad \bar{\beta})$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \iff \langle\psi| = \bar{\alpha}\langle 0| + \bar{\beta}\langle 1|$$

Zur Produktbildung von $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ und $|\phi\rangle = \begin{pmatrix} \gamma \\ \delta \end{pmatrix}$ ist das innere Produkt $\langle\psi|\phi\rangle$ (auch *bra-ket* genannt) eine Möglichkeit:
Skalarprodukt

$$\langle\psi|\phi\rangle = (\bar{\alpha} \quad \bar{\beta}) \cdot \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \bar{\alpha}\gamma + \bar{\beta}\delta.$$

Lineare Algebra

(Konjugieren von komplexen Zahlen) // $\overline{z_1 + z_2} = \overline{z_1} + \overline{z_2}$
 $\overline{z_1 z_2} = \overline{z_1} \cdot \overline{z_2}$

Für dieses gilt $\langle \phi | \psi \rangle = \overline{\langle \psi | \phi \rangle}$, denn

$$\langle \phi | \psi \rangle = \bar{\gamma}\alpha + \bar{\delta}\beta = \overline{\bar{\alpha}\gamma + \bar{\beta}\delta} = \overline{\langle \psi | \phi \rangle}.$$

Darüber hinaus ist das innere Produkt von $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ mit sich selbst

$$\langle \psi | \psi \rangle = |\alpha|^2 + |\beta|^2 = 1$$

In diesem Sinne kann $\langle \psi | \psi \rangle$ als Gesamtwahrscheinlichkeit interpretiert werden und $|\psi\rangle$ ist normalisiert, wenn $\langle \psi | \psi \rangle = 1$ gilt.

Lineare Algebra

Für das innere Produkt der verschiedenen Basen gilt

$$\langle 0|1\rangle = 0, \quad \langle +|-\rangle = 0, \quad \langle i|-i\rangle = 0.$$

Zustände (bzw. Vektoren) mit verschwindendem inneren Produkt nennen wir *orthogonal*.

Sind Zustände normalisiert und orthogonal, heißen sie *orthonormal*.

Also sind $|0\rangle, |1\rangle$ orthonormal, ebenso $|+\rangle, |-\rangle$ und $|i\rangle, |-i\rangle$.

Das innere Produkt kann auch verwendet werden um die Amplituden zu beschreiben bzw. zu bestimmen.

Lineare Algebra

Beispiel: Wir betrachten aus früherem Beispiel $|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$ und messen bzgl. der Z-Basis $\{|0\rangle, |1\rangle\}$. Die Amplitude von $|0\rangle$ ist

$$\langle 0|\psi\rangle = \langle 0| \left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle \right) = \frac{\sqrt{3}}{2}\langle 0|0\rangle + \frac{1}{2}\underbrace{\langle 0|1\rangle}_{\text{verschwindet, weil orthonormal}} = \frac{\sqrt{3}}{2}$$

und analog für $|1\rangle$ ist

$$\langle 1|\psi\rangle = \frac{\sqrt{3}}{2}\underbrace{\langle 1|0\rangle}_{\text{verschwindet, weil orthonormal}} + \frac{1}{2}\langle 1|1\rangle = \frac{1}{2}.$$

Die Messwahrscheinlichkeiten sind $|\frac{\sqrt{3}}{2}|^2 = \frac{3}{4}$ und $|\frac{1}{2}|^2 = \frac{1}{4}$. Ebenso erkennen wir

$$|\psi\rangle = \langle 0|\psi\rangle|0\rangle + \langle 1|\psi\rangle|1\rangle.$$

Lineare Algebra

Das war nur bedingt hilfreich und hätte auch durch ablesen festgestellt werden können.

Übung: Messen Sie $|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$ bzgl. der X-Basis $\{|+\rangle, |-\rangle\}$.

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ \langle +| &= \frac{1}{\sqrt{2}}(\langle 0| + \langle 1|) \\ &= \frac{1}{\sqrt{2}}\langle 0| + \frac{1}{\sqrt{2}}\langle 1| \end{aligned}$$

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

$$\langle -| = \frac{1}{\sqrt{2}}\langle 0| - \frac{1}{\sqrt{2}}\langle 1|$$

$$\begin{aligned} \langle +|\psi\rangle &= \langle +\left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle\right) \\ &= \frac{\sqrt{3}}{2}\langle +|0\rangle + \frac{1}{2}\langle +|1\rangle \\ &= \frac{\sqrt{3}}{2}\left(\frac{1}{\sqrt{2}}(\langle 0| + \langle 1|)|0\rangle + \right. \\ &\quad \left.\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(\langle 0| + \langle 1|)|1\rangle\right)\right) \\ &= \frac{\sqrt{3}}{2}\left(\frac{1}{2}\cdot\langle 0|0\rangle + \frac{1}{2}\cdot\langle 1|0\rangle + \right. \\ &\quad \left.\frac{1}{2}\cdot\frac{1}{\sqrt{2}}\cdot\langle 0|1\rangle + \frac{1}{2}\cdot\frac{1}{\sqrt{2}}\cdot\langle 1|1\rangle\right) \end{aligned}$$

Lineare Algebra

Das war nur bedingt hilfreich und hätte auch durch ablesen festgestellt werden können.

Übung: Messen Sie $|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$ bzgl. der X-Basis $\{|+\rangle, |-\rangle\}$.

Darstellung bzgl. innerem Produkt

Für eine Orthonormalbasis $\{|a\rangle, |b\rangle\}$ kann der Zustand eines Qubits $|\psi\rangle$ geschrieben werden als $|\psi\rangle = \alpha|a\rangle + \beta|b\rangle$ für $\alpha = \langle a|\psi\rangle$ und $\beta = \langle b|\psi\rangle$.

$\langle a|\psi\rangle$ heißt auch *Projektion* von $|\psi\rangle$ auf $|a\rangle$.

Operationen auf Qubits

Logische Gatter operieren auf Bits. Von einem *Quantengatter* erwarten wir eine Operation, die ein Qubits in ein (verändertes) Qubit überführt, insbesondere also die Nebenbedingung an die Amplituden respektiert.

Für $|0\rangle$ und $|1\rangle$ und ein Quantengatter U erwarten wir

$$U|0\rangle = a|0\rangle + b|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$$

$$U|1\rangle = c|0\rangle + d|1\rangle = \begin{pmatrix} c \\ d \end{pmatrix}$$

Operationen auf Qubits

Da wir Operationen auf einer Basis betrachten, resultiert

$$U = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

mit

$$U|0\rangle = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$$

$$U|1\rangle = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}$$

Operationen auf Qubits

Betrachten wir die Wirkung von U auf einem Qubit
 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, dann ist

$$\begin{aligned} U|\psi\rangle &= \begin{pmatrix} a & c \\ b & d \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a\alpha + c\beta \\ b\alpha + d\beta \end{pmatrix} \\ &= (a\alpha + c\beta)|0\rangle + (b\alpha + d\beta)|1\rangle \end{aligned}$$

Dies ist ein zulässiges Qubit, falls

$$|a\alpha + c\beta|^2 + |b\alpha + d\beta|^2 = 1$$

gilt. In Fall von Operationen auf Qubits läuft es auf *unitäre* Matrizen hinaus.²

²Wir verzichten hier auf die mathematische Herleitung.

Operationen auf Qubits

Grundlegende Gatter, die auf einem Qubit operieren, sind

Gatter	Matrix	Gatter	Matrix
Identität	$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	Phase S	$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
Pauli X	$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	T	$T = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(\frac{i\pi}{4}\right) \end{pmatrix}$
Pauli Y	$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Hadamard H	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Pauli Z	$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	-	-

Operationen auf Qubits

Allgemeiner: Rechenschritte auf Qubits

Definition (transponierte Matrix)

Sei

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$$

eine Matrix (mit komplexen Einträgen), dann heißt

$$A^T := \begin{pmatrix} a_{11} & \dots & a_{m1} \\ \vdots & & \vdots \\ a_{1n} & \dots & a_{mn} \end{pmatrix}$$

die *Transponierte* von A.

Operationen auf Qubits

Aus den Überlegungen zu Vektoren kennen wir im Prinzip schon

Definition (konjugierte und adjungierte Matrix)

Sei $A = (a_{ij}) \in \mathbb{C}^{m \times n}$. Die Matrix $\bar{A} := (\bar{a}_{ij}) \in \mathbb{C}^{m \times n}$ heißt die *Konjugierte* von A , und $A^\dagger := (\bar{A})^T$ die *Adjungierte* von A .

Definition (unitäre Matrix)

Eine Matrix $A = (a_{ij}) \in \mathbb{C}^{n \times n}$ heißt *unitär*, wenn $A^\dagger = A^{-1}$ gilt.

Es folgt, dass unitäre Matrizen / Quantengatter notwendig *invertierbar* sind, denn nach Definition gilt $A^\dagger A = AA^\dagger = I_n$. Insbesondere sind *reversible logische Gatter* auch zulässige Quantengatter.

Operationen auf Qubits

Wie schon bei der Operation eines Quantengatters auf einem einzelnen Qubit gesehen, wirkt ein Quantengatter (Matrix) durch Multiplikation von links auf einem Qubit (Vektor).

Erinnerung: Die Multiplikation eines Vektors mit einer (quadratischen) Matrix beschreibt eine lineare Abbildung.

In unserem Fall liefert die Multiplikation eines Vektors mit einer unitären Matrix $A \in \mathbb{C}^{n \times n}$ eine unitäre Transformation

$$A : \mathbb{C}^n \rightarrow \mathbb{C}^n, v \mapsto Av.$$

Operationen auf Qubits

Definition (Hadamard-Matrix)

Die Matrix

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

heißt *Hadamard-Matrix*.



Lemma

Die Hadamard-Matrix ist unitär.

Beweis: Übung.

Jacques Hadamard

Lösung nochmal ansehen !

Operationen auf Qubits

Wir untersuchen die Wirkung der Hadamard-Transformation auf der Z-Basis $\{|0\rangle, |1\rangle\}$. Wegen

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

gilt

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

Operationen auf Qubits

Analog:

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

Da $H = H^{-1}$ gilt, erhalten wir nach wiederholter Anwendung

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

und

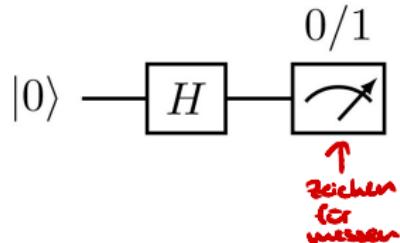
$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{H} |1\rangle.$$

Zufallszahlengenerator

Erste Anwendung: Ein (theoretisch echter) Zufallszahlengenerator

Algorithmus: Zufallszahlengenerator

1. $|x\rangle \leftarrow |0\rangle$
2. $|x\rangle \leftarrow H|x\rangle$
3. Messe $|x\rangle$



Analyse:

- Schritt 1: Qubit wird in den Anfangszustand $|0\rangle$ versetzt.
- Schritt 2: Anwenden der Hadamard-Transformation. Das Qubit befindet sich dann im Zustand $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
- Messen des Qubits liefert mit Wahrscheinlichkeit $1/2$ den Zustand $|0\rangle$ und mit Wahrscheinlichkeit $1/2$ den Zustand $|1\rangle$.

Zufallszahlengenerator

Übung: Ersetzen Sie die erste Zeile des Algorithmus durch

- $|x\rangle \leftarrow |1\rangle$, bzw.
- $|x\rangle \leftarrow \alpha|0\rangle + \beta|1\rangle$, mit $|\alpha|^2 + |\beta|^2 = 1$.

Welches Verhalten ergibt sich dann?

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

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

$$|\alpha|^2 = \left|\frac{1}{\sqrt{2}}\right|^2 \quad |\beta|^2 = \left|\frac{1}{\sqrt{2}}\right|^2$$

$$\underline{\underline{\alpha^2 = \frac{1}{2}}}$$

$$\underline{\underline{\beta^2 = \frac{1}{2}}}$$

1. Schritt: Qubit in zulässigen Zustand

2. Schritt: $\alpha|0\rangle + \beta|1\rangle$

$$\begin{aligned} & \alpha \cdot \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \beta \cdot \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{\alpha + \beta}{\sqrt{2}}|0\rangle + \frac{\alpha - \beta}{\sqrt{2}}|1\rangle \end{aligned}$$

Zufallszahlengenerator

Übung: Ersetzen Sie die erste Zeile des Algorithmus durch

- $|x\rangle \leftarrow |1\rangle$, bzw.
- $|x\rangle \leftarrow \alpha|0\rangle + \beta|1\rangle$, mit $|\alpha|^2 + |\beta|^2 = 1$.

Welches Verhalten ergibt sich dann?

Bitte beachten Sie

- In der Realität liefert die dafür notwendige Hardware keine gleichmäßige Werteverteilung.
- Die notwendige Nachbearbeitung ist nicht trivial.
- Kein aktuell von diversen Start-Ups vertriebener QRNG ist nach gängigen Kriterien zertifiziert.

Quantenregister

Ein Qubit ist für komplexere Anwendungen nicht ausreichend. Wir benötigen zunächst eine Möglichkeit mehrere Qubits zu beschreiben, um perspektivisch auch auf mehreren Qubits Berechnungen durchführen zu können.

Zur Beschreibung von *Quantenregistern* verwendet man das *Tensorprodukt*³.

³Formale Definition folgt später.

Quantenregister

Beispiel: Das (Tensor-)Produkt von $|0\rangle$ mit sich selbst ist $|0\rangle \otimes |0\rangle = |0\rangle|0\rangle = |00\rangle$ und durch kombinierte Produktbildung von $|0\rangle$ und $|1\rangle$ ergibt sich eine Basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$.

Ein allgemeiner Zustand bzgl. dieser Basis ist eine Superposition $\alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$ mit $|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1$.

Messen wir bzgl. dieses Basis, erhalten wir

- $|00\rangle$ mit Wahrscheinlichkeit $|\alpha_0|^2$.
- $|01\rangle$ mit Wahrscheinlichkeit $|\alpha_1|^2$.
- $|10\rangle$ mit Wahrscheinlichkeit $|\alpha_2|^2$.
- $|11\rangle$ mit Wahrscheinlichkeit $|\alpha_3|^2$.

Quantenregister

Allgemeiner: 2-Qubit Register

$R = |x_1\rangle|x_0\rangle$ mit $|x_0\rangle = \gamma_0|0\rangle + \gamma_1|1\rangle$ und $|x_1\rangle = \beta_0|0\rangle + \beta_1|1\rangle$.

Dann ist

$$\begin{aligned} R &= |x_1\rangle|x_0\rangle \\ &= \beta_0\gamma_0|0\rangle|0\rangle + \beta_0\gamma_1|0\rangle|1\rangle + \beta_1\gamma_0|1\rangle|0\rangle + \beta_1\gamma_1|1\rangle|1\rangle. \end{aligned}$$

Kurzschreibweise: $\alpha_{ij} = \beta_i\gamma_j$ und $|i\rangle|j\rangle = |ij\rangle$. Der Bitstring wird durch die in der Binärdarstellung repräsentierte Zahl ersetzt:

$$\begin{aligned} R &= \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \\ &= \alpha_0|0\rangle + \alpha_1|1\rangle + \alpha_2|2\rangle + \alpha_3|3\rangle. \end{aligned}$$

Quantenregister

Beobachtung: Die Amplituden $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ ergeben sich als Produkt der Amplituden der ursprünglichen Qubits
 $|x_0\rangle = \gamma_0|0\rangle + \gamma_1|1\rangle$ und $|x_1\rangle = \beta_0|0\rangle + \beta_1|1\rangle$.

Übung: Zeigen Sie: Aus $|\gamma_0|^2 + |\gamma_1|^2 = 1$ und $|\beta_0|^2 + |\beta_1|^2 = 1$ folgt $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$.

$$\begin{aligned} 1 = 1 \cdot 1 &= (|\beta_0|^2 + |\beta_1|^2)(|\gamma_0|^2 + |\gamma_1|^2) = \underbrace{|\beta_0\gamma_0|^2}_{= |\alpha_{00}|^2} + \underbrace{|\beta_0\gamma_1|^2}_{= |\alpha_{01}|^2} + \underbrace{|\beta_1\gamma_0|^2}_{= |\alpha_{10}|^2} + \underbrace{|\beta_1\gamma_1|^2}_{= |\alpha_{11}|^2} \\ &= |\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 \end{aligned}$$

Quantenregister

Beobachtung: Die Amplituden $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ ergeben sich als Produkt der Amplituden der ursprünglichen Qubits
 $|x_0\rangle = \gamma_0|0\rangle + \gamma_1|1\rangle$ und $|x_1\rangle = \beta_0|0\rangle + \beta_1|1\rangle$.

Übung: Zeigen Sie: Aus $|\gamma_0|^2 + |\gamma_1|^2 = 1$ und $|\beta_0|^2 + |\beta_1|^2 = 1$ folgt $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$.

Übung: Betrachten Sie das 2-Qubit Register $R = |x_1\rangle|x_0\rangle$ mit $|x_0\rangle = \frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle$ und $|x_1\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$. Bestimmen Sie die Amplituden $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ und führen Sie eine Messung bzgl. der (kombinierten) Basis durch.

Übung: Betrachten Sie das 2-Qubit Register $R = |x_1\rangle|x_0\rangle$ mit $|x_0\rangle = \frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle$ und $|x_1\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$. Bestimmen Sie die Amplituden $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ und führen Sie eine Messung bzgl. der (kombinierten) Basis durch.

$$\begin{aligned} R &= \frac{1}{2} \cdot \frac{1}{2} |0\rangle|0\rangle - \frac{1}{2} \cdot \frac{\sqrt{3}}{2} |0\rangle|1\rangle + \frac{\sqrt{3}}{2} \cdot \frac{1}{2} |1\rangle|0\rangle - \frac{\sqrt{3}}{2} \cdot \frac{\sqrt{3}}{2} |1\rangle|1\rangle \\ &= \frac{1}{4} |00\rangle - \frac{\sqrt{3}}{4} |01\rangle + \frac{\sqrt{3}}{4} |10\rangle - \frac{3}{4} |11\rangle \\ &= \frac{1}{4} |0\rangle - \frac{\sqrt{3}}{4} |1\rangle + \frac{\sqrt{3}}{4} |2\rangle - \frac{3}{4} |3\rangle \end{aligned}$$

Also:

$$\alpha_0 = \frac{1}{4}, \quad \alpha_1 = \frac{\sqrt{3}}{4}, \quad \alpha_2 = -\frac{\sqrt{3}}{4}, \quad \alpha_3 = \frac{3}{4}$$

und eine Messung führt

- mit Wk. $|\frac{1}{4}|^2 = \frac{1}{16}$ auf $|00\rangle$
- mit Wk. $|\frac{\sqrt{3}}{4}|^2 = \frac{3}{16}$ auf $|01\rangle$
- mit Wk. $|\frac{-\sqrt{3}}{4}|^2 = \frac{3}{16}$ auf $|10\rangle$
- mit Wk. $|\frac{3}{4}|^2 = \frac{9}{16}$ auf $|11\rangle$

Quantenregister

Nun für allgemeine Quantenregister:

Definition (Quantenregister)

Ein *Quantenregister R* der Länge $n \geq 1$ hat die Form

$R = |x_{n-1}\rangle|x_{n-2}\rangle\dots|x_0\rangle = |x_{n-1}x_{n-2}\dots x_0\rangle$. Es kann sich in einem Zustand der Form

$$\sum_{i=0}^{2^n-1} \alpha_i |i\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n-1} |2^n - 1\rangle$$

befinden, wobei $|i\rangle = |\text{bin}(i)\rangle$ und $\alpha_i \in \mathbb{C}$ für $i = 0, \dots, 2^n - 1$ gelte.
Es gilt $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$ und beim Messen des Quantenregisters beobachtet man den Zustand $|i\rangle$ mit Wahrscheinlichkeit $|\alpha_i|^2$.

Quantenregister

Der Inhalt eines (logischen) n -Bit Registers ist ein n -Bit String, d.h. es sind 2^n Inhalte möglich. Ein n -Qubit Register befindet sich in Superposition all dieser Zustände.

Für $n = 300$ Qubits müssten $2^{300} \approx 2,04 \cdot 10^{90}$ Amplituden berücksichtigt werden. Die Anzahl der Atome im sichtbaren Universum wird auf 10^{78} bis 10^{82} geschätzt. Das ist ein Hinweis darauf, dass es schwierig ist für einen klassischen Computer einen Quantencomputer zu simulieren.

Quantenregister

Erinnerung: Ein Qubit ist ein Vektor im zweidimensionalen Raum.

Die Zustände eines Quantenregisters mit n Qubits entsprechen Vektoren in einem 2^n -dimensionalen komplexen Vektorraum. Die Basis bilden die einzelnen Komponenten der Superposition, also

Basiszustände → $|0\dots00\rangle, |0\dots01\rangle, \dots, |1\dots11\rangle$.

Beispiel: Für ein 2-Qubit Register ist $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ eine Basis mit der entsprechenden Zuordnung

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Quantenregister

Da die Zustandsvektoren eines n -Bit Quantenregisters je 2^n Komponenten besitzen, entsprechen die einzelnen Rechenschritte eines Quantencomputers unitären Transformationen, die durch unitäre $2^n \times 2^n$ -Matrizen darstellbar sind.

Im Fall von Ein-Qubit Systemen haben wir spezielle Matrizen bzw. Transformationen kennengelernt, die auf diesen operieren.

Es gibt auch Quantengatter die auf zwei Qubits gleichzeitig operieren. Einige dieser Gatter wollen wir näher betrachten (auch als Vorbereitung für weitere Quantenalgorithmen).

Quantenregister

Das **CNOT-Gatter** bzw. ***controlled-NOT* Gatter** invertiert das rechte Qubits, wenn das linke (Kontroll-)Qubit auf 1 gesetzt ist, d.h.

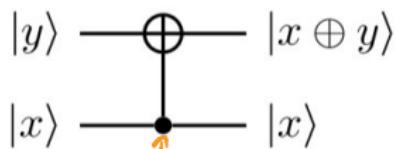
Controllbit (wenn Controllbit auf 1,
rechtes bit flippen)

$$\text{CNOT } |00\rangle = |00\rangle$$

$$\text{CNOT } |01\rangle = |01\rangle$$

$$\text{CNOT } |10\rangle = |11\rangle$$

$$\text{CNOT } |11\rangle = |10\rangle$$



wenn ausgewählt rechtes
flippen, wenn nicht ausgewählt
linkes flippen

$$A_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

oder kompakt

$$\text{CNOT} : |x, y\rangle \mapsto |x, x \oplus y\rangle.$$

Quantenregister

Notation: In Schaltungen wird absteigend sortiert, d.h. das am weitesten rechts geschriebene Qubit im Register steht in der Schaltung am weitesten oben und das am weitesten links stehende Qubit steht in der Schaltung unten.

Übung:

1. Leiten Sie die Matrixdarstellung A_{CNOT} von CNOT her.
2. Zeigen Sie, dass A_{CNOT} unitär ist.

Erinnerung: Die Multiplikation eines Vektors mit einer (unitären) Matrix $A \in \mathbb{C}^{n \times n}$ liefert eine (unitäre) Transformation $A : \mathbb{C}^n \rightarrow \mathbb{C}^n, v \mapsto Av$. In den Spalten von A stehen die Bilder der Standard-Basisvektoren bei der Abbildung $v \mapsto Av$.

Übung:

- Leiten Sie die Matrixdarstellung A_{CNOT} von CNOT her.
- Zeigen Sie, dass A_{CNOT} unitär ist.

$$A_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$1) \quad \text{CNOT } |00\rangle = |00\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\text{CNOT } |01\rangle = |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \left. \right\} \text{also: } A_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\text{CNOT } |10\rangle = |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\text{CNOT } |11\rangle = |10\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$2) \quad A_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}; \quad A_{\text{CNOT}}^{\dagger} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad // A_{\text{CNOT}} = A_{\text{CNOT}}^{\dagger}$$

$$A_{\text{CNOT}} \cdot A_{\text{CNOT}}^{\dagger} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = I_4$$

\rightarrow unitär

Quantenregister

Es gilt noch mehr: A_{CNOT} ist eine Permutationsmatrix.

Erinnerung: Sei $\pi \in S_n$ (S_n die symmetrische Gruppe; deren Elemente nennt man Permutationen). Dann ist die dazugehörige $(n \times n)$ -Permutationsmatrix $P_\pi = (p_{ij})$ definiert durch $p_{ij} = \delta_{\pi(i), j}$.
Per Definition ist P_π also eine quadratische Matrix, die in jeder Zeile und jeder Spalte genau einen Eintrag 1 und sonst nur Nullen enthält

Übung: Zeigen Sie: Permutationsmatrizen sind unitär.

Konkret
Funktion

$$\delta = \begin{cases} 1, & i=j \\ 0, & \text{sonst} \end{cases}$$

Übung: Zeigen Sie: Permutationsmatrizen sind unitär.

Erinnerung: Sei $\pi \in \text{Sn}$. Die zugehörige $(n \times n)$ -Permutationsmatrix $P_\pi = (p_{ij})$ durch:

$$p_{ij} = \delta_{\pi(i), j} = \begin{cases} 1 & \text{falls } \pi(i) = j, \\ 0 & \text{sonst} \end{cases}$$

Per Definition sind die Einträge von P_π Elemente aus $\{0, 1\} \subset \mathbb{Z}$, also gilt $\overline{P_\pi} = P_\pi$.

Transponieren liefert nun $P_\pi^T = (p_{ij}) = (\delta_{\pi(i), j})$ mit $\delta_{\pi(i), j} = \begin{cases} 1 & \text{falls } \pi(j) = i \\ 0 & \text{sonst} \end{cases}$

$$= P_{\pi^{-1}} = P_\pi^{-1}$$

Wobei π^{-1} die zu π inverse Permutation bezgl. Komposition bezeichnet.
Mit $P_\pi^T = P_{\pi^{-1}}$ folgt $P_\pi^T \cdot (P_\pi)^T = P_\pi^T = P_\pi^{-1}$

→ Permutationsmatrizen sind also unitär.

Quantenregister

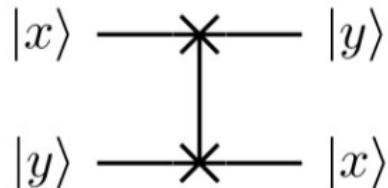
Das **SWAP-Gatter** vertauscht zwei Qubits, d.h.

$$\text{SWAP } |00\rangle = |00\rangle$$

$$\text{SWAP } |01\rangle = |10\rangle$$

$$\text{SWAP } |10\rangle = |01\rangle$$

$$\text{SWAP } |11\rangle = |11\rangle$$



oder kompakt

$$\text{SWAP} : |x\rangle|y\rangle \mapsto |y\rangle|x\rangle.$$

$$A_{\text{SWAP}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Quantenregister

Soll ein bel. 2-Qubit Gatter U mit einer Kontrolle versehen werden, wird daraus das ***CU-Gatter*** oder auch *controlled-U* Gatter. Dabei wird U auf das rechte Qubit angewendet, wenn das Kontrollqubit auf 1 gesetzt ist:

$$CU |00\rangle = |00\rangle$$

$$CU |01\rangle = |01\rangle$$

$$CU |10\rangle = |1\rangle \otimes U|0\rangle$$

$$CU |11\rangle = |1\rangle \otimes U|1\rangle.$$

Quantenregister

Für

$$U|0\rangle = a|0\rangle + b|1\rangle$$

$$U|1\rangle = c|0\rangle + d|1\rangle$$

folgt

$$A_{CU} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{pmatrix}$$

Quantenregister

Unter diesem Aspekt gilt

$$CX = CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

und natürlich sind weitere Konstruktionen spezieller 2-Qubit Gatter denkbar.

Mit 2-Qubit Gattern können wir den (historisch) ältesten Quantenalgorithmus untersuchen.

Algorithmus von Deutsch

Problem von Deutsch (nach David Deutsch, geb. 1953)

Wir wollen die *Parität* $b_0 \oplus b_1$ zweier unbekannter Bits b_0 und b_1 bestimmen. Äquivalent dazu ist die Frage, ob die Anzahl der Einsen *gerade* oder *ungerade* ist.

Klassische Analogie: Eine echte Münze (Kopf und Zahl) soll von einer gefälschten Münze (beide Seiten Kopf) unterschieden werden. Die Münze muss zweimal betrachtet werden, je einmal von jeder Seite.

Bietet uns ein Quantencomputer in einer solchen (oder ähnlichen) Situation Vorteile?

Algorithmus von Deutsch

Abstrakt: Gegeben eine Funktion $f : \{0, 1\} \rightarrow \{0, 1\}$ und es gibt ein *Orakel* das uns zu einem Bit $b \in \{0, 1\}$ den Wert $f(b)$ liefert. Das Orakel sagt immer die Wahrheit. Wir können es als eine *Subroutine* auffassen.

Die Funktion f heißt *konstant*, wenn $f(0) = f(1)$ gilt. Im Fall $f(0) \neq f(1)$ heißt f *balanciert*.

Frage: Ist f konstant oder balanciert?

Klassisch: Zwei Anfragen an das Orakel, nämlich $f(0)$ und $f(1)$.

Idee QC: Versetze ein Qubit in eine Superposition über die möglichen Eingaben 0 und 1 von f .

Algorithmus von Deutsch

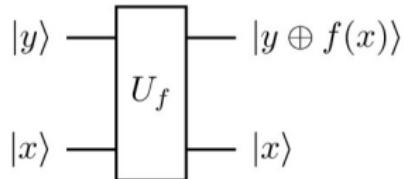
Achtung: f ist möglicherweise nicht umkehrbar (f konstant).
Rechenschritte auf Quantencomputern müssen aber umkehrbar sein.

Wir verwenden

$$U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle.$$

Das extra Qubit $|y\rangle$ nennt man auch *Antwortqubit* oder *target Qubit*;
 $|x\rangle$ nennt man *input Qubit*.
Das Quantenorakel können wir als eine Quanten-Subroutine verstehen.

Übung: Zeige, das U_f unitär ist.



Übung: Zeige, das U_f unitär ist.

1. Möglichkeit: f ist die Nullfunktion, d.h. $f: \{0,1\} \rightarrow \{0,1\}$ mit $f(0)=0, f(1)=0$

Dann gilt: $U_f: \begin{cases} |00\rangle \mapsto |0,0 \oplus f(0)\rangle = |00\rangle \\ |01\rangle \mapsto |0,1 \oplus f(0)\rangle = |01\rangle \\ |10\rangle \mapsto |1,0 \oplus f(1)\rangle = |10\rangle \\ |11\rangle \mapsto |1,1 \oplus f(1)\rangle = |11\rangle \end{cases}$

\Rightarrow Das heißt U_f ist die Identität und wird durch I_4 beschrieben. $U_f = I_4$ ist unitär.

2. Möglichkeit: f ist die Identität, d.h. $f: \{0,1\} \rightarrow \{0,1\}$ mit $f(0)=0, f(1)=1$

Dann gilt: $U_f: \begin{cases} |00\rangle \mapsto |0,0 \oplus f(0)\rangle = |00\rangle \\ |01\rangle \mapsto |0,1 \oplus f(0)\rangle = |01\rangle \\ |10\rangle \mapsto |1,0 \oplus f(1)\rangle = |11\rangle \\ |11\rangle \mapsto |1,1 \oplus f(1)\rangle = |10\rangle \end{cases}$

\Rightarrow Die Abbildung kennen wir bereits unter CNOT, d.h. $U_f \cdot \text{CNOT} \Rightarrow U_f$ ist unitär.

3. Möglichkeit: f ist die Negation, d.h. $f: \{0,1\} \rightarrow \{0,1\}$, mit $f(0)=1, f(1)=0$

Dann gilt: $U_f: \begin{cases} |00\rangle \mapsto |0,0 \oplus f(0)\rangle = |01\rangle \\ |01\rangle \mapsto |0,1 \oplus f(0)\rangle = |10\rangle \\ |10\rangle \mapsto |1,0 \oplus f(1)\rangle = |10\rangle \\ |11\rangle \mapsto |1,1 \oplus f(1)\rangle = |11\rangle \end{cases}$

Wir wissen: U_f kann durch eine Matrix beschrieben werden.

Die mit $|i\rangle$ beschriftete Spalte beschreibt das Bild des Basisvektors $|i\rangle$

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

$\Rightarrow U_f$ ist eine Permutationsmatrix, also unitär.

4. Möglichkeit: f ist die Einheitsfunktion, also $f: \{0,1\} \rightarrow \{0,1\}, f(0)=1, f(1)=1$

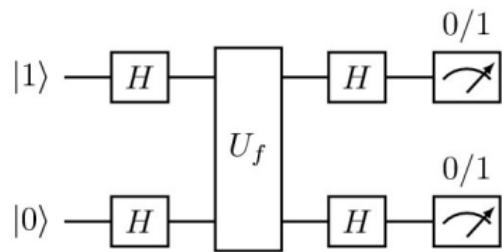
Dann gilt: $\begin{cases} |00\rangle \mapsto |0,0 \oplus f(0)\rangle = |01\rangle \\ |01\rangle \mapsto |0,1 \oplus f(0)\rangle = |00\rangle \\ |10\rangle \mapsto |1,0 \oplus f(1)\rangle = |11\rangle \\ |11\rangle \mapsto |1,1 \oplus f(1)\rangle = |10\rangle \end{cases}$ $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

$\Rightarrow U_f$ ist eine Permutationsmatrix und als solche unitär.

Algorithmus von Deutsch

Algorithmus von Deutsch

1. $|x\rangle|y\rangle \leftarrow |0\rangle|1\rangle$
2. $|x\rangle|y\rangle \leftarrow H|x\rangle H|y\rangle$
3. $|x\rangle|y\rangle \leftarrow U_f |x\rangle|y\rangle$
4. $|x\rangle|y\rangle \leftarrow H|x\rangle H|y\rangle$
5. Messe das Register $|x\rangle|y\rangle$:
 - $|0\rangle|1\rangle$: f ist konstant
 - $|1\rangle|1\rangle$: f ist balanciert



Algorithmus von Deutsch

Analyse des Algorithmus: In Schritt 2 wird $|x\rangle|y\rangle$ durch Hadamard-Transformation auf $|0\rangle|1\rangle$ in

$$\begin{aligned} |\phi_2\rangle &= H|0\rangle H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \cdot \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ &= \frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle) \end{aligned}$$

überführt. Das ist eine Superposition über alle Basiszustände des Registers.

Algorithmus von Deutsch

In Schritt 3 wird $U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$ angewandt:

$$\begin{aligned} |\phi_3\rangle &= U_f |\phi_2\rangle = \frac{1}{2} (U_f |0\rangle|0\rangle - U_f |0\rangle|1\rangle + U_f |1\rangle|0\rangle - U_f |1\rangle|1\rangle) \\ &= \frac{1}{2} (|0\rangle|f(0)\rangle - |0\rangle|1 \oplus f(0)\rangle + |1\rangle|f(1)\rangle - |1\rangle|1 \oplus f(1)\rangle) \\ &= \frac{1}{2} (|0\rangle(|f(0)\rangle - |1 \oplus f(0)\rangle) + |1\rangle(|f(1)\rangle - |1 \oplus f(1)\rangle)). \end{aligned}$$

Wir können die Terme weiter vereinfachen...

Übung: Für $x \in \{0, 1\}$ ist $|f(x)\rangle - |1 \oplus f(x)\rangle = (-1)^{f(x)}(|0\rangle - |1\rangle)$.

$$|f(x)\rangle - |1 \oplus f(x)\rangle = \begin{cases} |0\rangle - |1 \oplus 0\rangle, & \text{for } f(x)=0 \\ |1\rangle - |1 \oplus 1\rangle, & \text{for } f(x)=1 \end{cases}$$

$$=(-1)^{f(x)}(|0\rangle - |1\rangle)$$

Algorithmus von Deutsch

... und erhalten

$$\begin{aligned} |\phi_3\rangle &= \frac{1}{2} \left((-1)^{f(0)} |0\rangle (|0\rangle - |1\rangle) + (-1)^{f(1)} |1\rangle (|0\rangle - |1\rangle) \right) \\ &= \frac{1}{2} \left((-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle \right) (|0\rangle - |1\rangle). \end{aligned}$$

Der Funktionswert wurde in das Vorzeichen der Amplituden des ersten Bits $|x\rangle$ von $|\phi_3\rangle$ verlagert, wobei

$$|x\rangle = \frac{1}{\sqrt{2}} \left((-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle \right).$$

Algorithmus von Deutsch

In **Schritt 4** erfolgt die Fallunterscheidung für die Funktion f :

1. Möglichkeit: f ist konstant, also $f(0) = f(1)$. Dann ist $(-1)^{f(0)} = (-1)^{f(1)}$ und entweder

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

für $f(0) = f(1) = 0$ oder

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

für $f(0) = f(1) = 1$.

Algorithmus von Deutsch

Für das zweite Qubit in $|\phi_3\rangle$ gilt

$$\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \xrightarrow{H} |1\rangle.$$

Damit enthält das Register $\pm|0\rangle|1\rangle$.

2. Möglichkeit: f ist balanciert, also $f(0) \neq f(1)$. Dann ist

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

für $f(0) = 0, f(1) = 1$;

Algorithmus von Deutsch

oder

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

für $f(0) = 1, f(1) = 0$. Das Register enthält dann $\pm|1\rangle|1\rangle$.

Damit wird im Fall f konstant $|0\rangle|1\rangle$ und im Fall f balanciert $|1\rangle|1\rangle$ gemessen.

Übung: Vollziehen Sie die Überlegungen für die Fälle

- $f(0) = 1, f(1) = 0$, d.h. f ist die Negation, und
- $f(0) = f(1) = 1$, d.h. f ist die 1-Funktion

nach.

Quantenregister

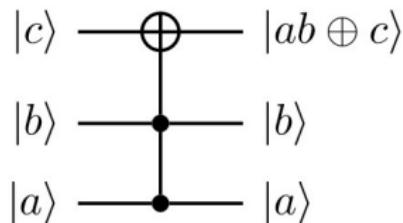
Für den Algorithmus von Deutsch war unser bisheriges Wissen über Ein-Qubit und 2-Qubit Systeme, sowie Operationen auf diesen, gerade ausreichend. Erwartungsgemäß kommt man auch mit 2-Qubit Systemen noch nicht sehr weit.

Tatsächlich gibt es auch spezielle Gatter die gleichzeitig auf drei Qubits arbeiten.

Das Toffoli Gatter

Toffoli $|a\rangle|b\rangle|c\rangle \mapsto |a\rangle|b\rangle|ab \oplus c\rangle$

ist ein solches Gatter.



Quantenregister

Das Toffoli Gatter ist

- reversibel
- und sogar *universell* für klassische Computer,

d.h. jeder effiziente klassische Algorithmus kann in einen effizienten Algorithmus der nur Toffoli Gatter verwendet, konvertiert werden.

Da das Toffoli Gatter auch ein Quantengatter ist, können Quantencomputer effizient berechnen, was auch klassische Computer effizient berechnen können. Anders:

$$P \subset BQP.$$

Tensorprodukt

Auf Dauer ist unsere bisherige Arbeitsweise (Steigerung in der Anzahl der Qubits führt zu (teils) grundlegend neuen Gattern) kein praktikabler Weg.

Wir benötigen eine *konstruktive* Methode, auch um bereits bekannte Gatter in einem erweiterten Setting verwenden zu können.

Tensorprodukt

Auf Dauer ist unsere bisherige Arbeitsweise (Steigerung in der Anzahl der Qubits führt zu (teils) grundlegend neuen Gattern) kein praktikabler Weg.

Wir benötigen eine *konstruktive* Methode, auch um bereits bekannte Gatter in einem erweiterten Setting verwenden zu können.

Vorgehen: Wir überlegen uns wie wir formal Register aus einzelnen Qubits zusammensetzen. Damit wollen wir Operationen auf einem Register durch Operationen auf einzelnen Qubits zusammenfügen.

Erinnerung: Qubits sind Linearkombinationen von Basisvektoren.

Tensorprodukt

Definition (Tensorprodukt von Vektorräumen - Teil 1)

Sei V_1 ein \mathbb{C} -Vektorraum mit Basis $\{e_0, \dots, e_{m-1}\}$ und V_2 ein \mathbb{C} -Vektorraum mit Basis $\{f_0, \dots, f_{n-1}\}$. Das *Tensorprodukt* $V_1 \otimes V_2$ dieser Räume ist ein mn -dimensionaler Vektorraum, dessen Basisvektoren mit

$$\begin{array}{cccc} e_0 \otimes f_0 & e_0 \otimes f_1 & \dots & e_0 \otimes f_{n-1} \\ e_1 \otimes f_0 & e_1 \otimes f_1 & \dots & e_1 \otimes f_{n-1} \\ \vdots & \vdots & & \vdots \\ e_{m-1} \otimes f_0 & e_{m-1} \otimes f_1 & \dots & e_{m-1} \otimes f_{n-1} \end{array}$$

bezeichnet werden.

Tensorprodukt

Definition (Tensorprodukt von Vektorräumen - Teil 2)

Ist

$$v_1 = \alpha_0 e_0 + \dots + \alpha_{m-1} e_{m-1} \in V_1$$

und

$$v_2 = \beta_0 f_0 + \dots + \beta_{n-1} f_{n-1} \in V_2,$$

dann ist ihr Tensorprodukt

$$v_1 \otimes v_2 = \left(\sum_{i=0}^{m-1} \alpha_i e_i \right) \otimes \left(\sum_{j=0}^{n-1} \beta_j f_j \right) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \alpha_i \beta_j (e_i \otimes f_j).$$

Tensorprodukt

Beispiel: Für $m = n = 2$ ist

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix}.$$

Also ist

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

und damit (wie bisher) $|1\rangle \otimes |0\rangle = |10\rangle$. Analog zeigt man $|0\rangle \otimes |0\rangle = |00\rangle$, $|0\rangle \otimes |1\rangle = |01\rangle$ und $|1\rangle \otimes |1\rangle = |11\rangle$.

Tensorprodukt

Allgemeiner liefert die Definition des Tensorproduktes für $|\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ und $|\psi\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, dass

$$|\phi\rangle \otimes |\psi\rangle = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle.$$

Das ergibt sich (wie bisher) auch durch „Ausmultiplizieren“ von

$$(\alpha_0|0\rangle + \alpha_1|1\rangle) \cdot (\beta_0|0\rangle + \beta_1|1\rangle).$$

Übung: Berechnen Sie

$$\begin{aligned} & \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \\ &= \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \cdot \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ &= \frac{1}{2}(|100\rangle - |101\rangle + |10\rangle - |11\rangle) \end{aligned}$$

Tensorprodukt

Lemma (Produkt von Zuständen)

Die Beschreibung eines Registers aus m Bits lässt sich aus dem m -fachen Tensorprodukt der Beschreibung eines Bits erzeugen.

Sind die Bits $|x_1\rangle, \dots, |x_m\rangle$ in den Zuständen $|\phi_1\rangle, \dots, |\phi_m\rangle$, so befindet sich das Register $|x_1 \dots x_m\rangle$ im Zustand $|\phi_1\rangle \otimes \dots \otimes |\phi_m\rangle$.

Die Amplituden können (wie bisher) durch Ausmultiplizieren berechnet werden (weshalb wir häufig \cdot statt \otimes schreiben werden, auch wenn es unpräzise ist).

Tensorprodukt

Wenn (einige) Quantenzustände als Tensorprodukt entstehen, liegt es nahe auch nach der *Faktorisierung* von Zuständen zu fragen. So ist

$$\frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \underbrace{\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)}_{|+\rangle} \otimes \underbrace{\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)}_{|-\rangle}$$
$$= |+\rangle \otimes |-\rangle$$
$$= |+\rangle |-\rangle.$$

Solche faktorisierbaren Zustände nennt man *Produktzustände* oder *separabel*.

Tensorprodukt

Andererseits gibt es auch Zustände die sich nicht in ein Tensorprodukt zerlegen lassen, etwa

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

Solche Zustände nennen wir *verschränkt*. Wir später diese später weiter untersuchen und lernen wie man dies nutzen kann.

Momentan sind wir an einer konstruktiven Methode zur Generierung von Gattern interessiert.

Tensorprodukt

Definition (Tensorprodukt von Matrizen)

Seien A und B Matrizen mit komplexen Einträgen, wobei

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m-1} & \dots & a_{mn} \end{pmatrix}.$$

Das *Tensorprodukt* $A \otimes B$ von A und B ist

$$A \otimes B = \begin{pmatrix} a_{11} \cdot B & \dots & a_{1n} \cdot B \\ \vdots & & \vdots \\ a_{m-1} \cdot B & \dots & a_{mn} \cdot B \end{pmatrix}.$$

Tensorprodukt

Beispiel:

$$I_2 \otimes I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes I_2 = \begin{pmatrix} I_2 & 0 \\ 0 & I_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = I_4$$

Als Verallgemeinerung des Hadamard Gatters ergibt sich

Definition (H_n)

Die 2^n -dimensionale Hadamard-Transformation H_n ist definiert durch

$$H_n = \bigotimes_{i=1}^n H.$$

Tensorprodukt

Übung: Berechnen Sie H_2 .

Wir beobachten, dass die folgenden Aktionen auf dasselbe Resultat führen

- Anwendung der durch die Matrizen A_1, \dots, A_m beschriebenen Transformationen auf die Bits $|x_1\rangle, \dots, |x_m\rangle$. Dabei wird jeweils A_i auf $|x_i\rangle$ angewandt.
- Anwendung der Transformation $A_1 \otimes \dots \otimes A_m$ auf das Register $|x_1\dots x_m\rangle$.

Mit dem Tensorprodukt haben wir also die Möglichkeit, Operationen auf Registern zu konstruieren / auszuführen.

Berechnen $S \otimes H_2$

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

$$H_2 = H \otimes H$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{bzw. } \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} = \dots$$

$$= \left(\boxed{\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}} \right) \otimes \left(\boxed{\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}} \right)$$

$$\left(\begin{array}{cccc} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{array} \right) = \frac{1}{2} \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}}$$

Tensorprodukt

Beispiel: Betrachte $R = |x\rangle|y\rangle$

- $H \otimes H = H_2$ beschreibt auf Registerebene die Anwendung von H auf $|x\rangle$ und auf $|y\rangle$
- $H \otimes I_2$ beschreibt die Anwendung von H auf $|x\rangle$ und lässt $|y\rangle$ unverändert

Übung: Berechnen Sie $H \otimes I_2$ und $I_2 \otimes H$. Ist das Tensorprodukt kommutativ?

Universelle Gatter

Für logische Gatter ist NAND universell, da mit diesem jedes andere logische Gatter konstruiert werden kann.

Eine Menge von Quantengattern ist eine **universelle Gattermenge**, wenn damit jedes Quantengatter mit beliebiger Genauigkeit approximiert werden kann.

Einige Besonderheiten sind das mit universellen Quantengattern die Möglichkeit gegeben sein muss

- Superposition zu erzeugen,
- Verschränkung herzustellen,
- Zustände mit komplexe Amplituden zu generieren,
- mehr als die **Clifford Gatter** {CNOT, H, S} enthalten sein muss.

Universelle Gatter

Unter den Clifford Gattern erzeugt H Superposition, CNOT Verschränkung und S liefert komplexwertige Amplituden. Nachweislich bilden diese Gatter aber keine universelle Gattermenge.

Nach dem **Satz von Gottesmann-Knill** können Quantenschaltkreise die nur aus den Clifford Gattern bestehen, von einem klassischen Computer effizient simuliert werden. Solche Schaltungen sind demnach nicht mächtiger als klassische Schaltkreise.

Universelle Gatter

Beispiele universeller Mengen für Quantencomputer sind

- $\{H, \text{alle Ein-Qubit Gatter}\}$
- $\{\text{CNOT}, H, T\}$, d.h. Austausch von S durch T liefert eine universelle Menge
- $\{\text{CNOT}, R_{\frac{\pi}{8}}, S\}$, d.h. Austausch von H durch $R_{\frac{\pi}{8}}$ liefert eine universelle Menge für

$$R_{\frac{\pi}{8}} = \begin{pmatrix} \cos\left(\frac{\pi}{8}\right) & -\sin\left(\frac{\pi}{8}\right) \\ \sin\left(\frac{\pi}{8}\right) & \cos\left(\frac{\pi}{8}\right) \end{pmatrix}$$

- $\{\text{Toffoli}, H, S\}$

Universelle Gatter

Der **Satz von Solovay-Kitaev** besagt, dass ein Quantengatter auf n Qubit durch eine universelle Menge von Quantengattern zu einer Präzision ε durch $\Theta(2^n \log^c(\frac{1}{\varepsilon}))$ Gatter approximiert werden kann; c eine Konstante.

Die Abhängigkeit von 2^n ist zu erwarten, da für die Operation auf n Qubits eine $2^n \times 2^n$ Matrix benötigt wird. Die Abhängigkeit von der Approximationsgüte ε in $\log^c(\frac{1}{\varepsilon})$ ist gut, da diese erwartbar klein sein soll und damit $\log^c(\frac{1}{\varepsilon})$ von der Größenordnung *polylog* ist. Die Approximation durch die universellen Gatter konvergiert damit schnell gegen das zu approximierende Gatter.

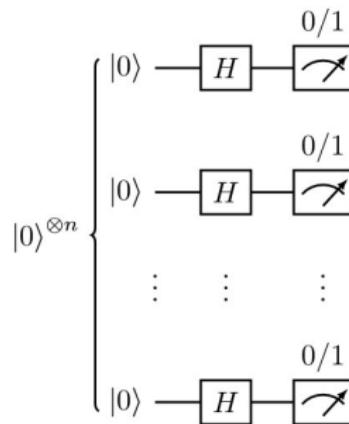
Erweiterter Zufallszahlengenerator

Mit den neuen Möglichkeiten können wir den Ansatz zur Erzeugung von Zufallszahlen erweitern:

Algorithmus: *n*-Bit Zufallsgenerator

Ausgabe: Zufallszahl zwischen 0 und $2^n - 1$

1. $R = |x_{n-1} \dots x_0\rangle \leftarrow |0 \dots 0\rangle$
2. $R \leftarrow H_n R$
3. Messe R



Erweiterter Zufallszahlengenerator

Analyse des Algorithmus: In **Schritt 2** wird H_n auf das Register angewandt; d.h. H wirkt auf jedes einzelne Bit, sodass

$$|0\rangle \dots |0\rangle \xrightarrow{H_n} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \dots \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle).$$

Für das Produkt zweier Faktoren beobachten wir

$$\begin{aligned} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \cdot \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) &= \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\ &= \frac{1}{2} (|0\rangle + |1\rangle + |2\rangle + |3\rangle). \end{aligned}$$

Erweiterter Zufallszahlengenerator

Insgesamt liefert die Produktbildung

$$\frac{1}{\sqrt{2^n}} (|0\dots00\rangle + |0\dots01\rangle + \dots + |1\dots1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle.$$

Bei der Messung in **Schritt 3** wird jeder Basiszustand des Registers mit Wahrscheinlichkeit $(1/\sqrt{2^n})^2 = 1/2^n$ angenommen. Jeder dieser Basiszustände repräsentiert eine der Zahlen 0 bis $2^n - 1$.

Messen von Registern

Um weitere Algorithmen betrachten zu können, müssen wir auch unser bisheriges Verständnis einer Messung verfeinern.

Unser bisheriges Verständnis der Messung bzgl. einer Basis hat sich auf die gesamte Basis bezogen.

Erinnerung: Sei R ein Register aus n Quantenbits, das sich im Zustand $|\phi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$ befindet. Die orthogonalen Vektoren $|0'\rangle, |1'\rangle, \dots, |(2^n - 1)'\rangle$ der Länge 1 seien die Messbasis von $|\phi\rangle$, d.h.

$$|\phi\rangle = \sum_{i=0}^{2^n-1} \alpha'_i |i'\rangle.$$

Dann befindet sich das Register nach Messung mit Wahrscheinlichkeit $|\alpha'_i|^2$ im Zustand $|i'\rangle$.

Messen von Registern

Es können auch einzelne Bits eines Registers gemessen werden:

Für $R = |xy\rangle$ im Zustand

$$|\phi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

kann bspw. das erste Bit gemessen werden. Das Ergebnis ist $|0\rangle$ oder $|1\rangle$.

1. Fall: Messen nach $|x\rangle = |0\rangle$

Das Register geht in eine Superposition von $|00\rangle$ und $|01\rangle$, genauer,

$$|\phi'\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$$

über. Die Wahrscheinlichkeit dafür beträgt $|\alpha_{00}|^2 + |\alpha_{01}|^2$.

Messen von Registern

2. Fall: Messen nach $|x\rangle = |1\rangle$

Das Register geht in eine Superposition von $|10\rangle$ und $|11\rangle$, genauer, den Zustand

$$|\phi'\rangle = \frac{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}$$

über. Die Wahrscheinlichkeit dafür beträgt $|\alpha_{10}|^2 + |\alpha_{11}|^2$.

Um einen zulässigen Quantenzustand zu erhalten, musste *normiert* werden.

Hier ist durch Messung ein Übergang von einer Superposition in eine andere Superposition (Folgezustand) entstanden. Im Vergleich dazu hat eine Messung bisher zu einer Auflösung der Superposition geführt.

Messen von Registern

Allgemein gilt:

Ist R ein Register aus n Quantenbits im Zustand $|\phi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$ und für $j \in \{1, \dots, n\}$ sei

$$I_{0,j} = \{i \in \{0, \dots, 2^n - 1\} : j\text{-tes Bit von links in } \text{bin}(i) \text{ von } i \text{ ist } |0\rangle\},$$
$$I_{1,j} = \{i \in \{0, \dots, 2^n - 1\} : j\text{-tes Bit von links in } \text{bin}(i) \text{ von } i \text{ ist } |1\rangle\}.$$

Wird das j -te Bit des Registers gemessen, so nimmt es mit

Wahrscheinlichkeit $\sum_{i \in I_{j,0}} |\alpha_i|^2$ den Wert $|0\rangle$ an. Das Register ist dann im Zustand

$$\frac{\sum_{i \in I_{j,0}} \alpha_i |i\rangle}{\sqrt{\sum_{i \in I_{j,0}} |\alpha_i|^2}}.$$

(Beachte: Alle $|i\rangle$ die hier auftreten, haben an Position j eine $|0\rangle$.)

Messen von Registern

Wird das j-te Bit des Registers gemessen, so nimmt es mit Wahrscheinlichkeit $\sum_{i \in I_{j,1}} |\alpha_i|^2$ den Wert $|1\rangle$ an. Das Register ist dann im Zustand

$$\frac{\sum_{i \in I_{j,1}} \alpha_i |i\rangle}{\sqrt{\sum_{i \in I_{j,1}} |\alpha_i|^2}}.$$

(Beachte: Alle $|i\rangle$ die hier auftreten, haben an Position j eine $|1\rangle$.)

Übung: Das 3-Qubit Register R sei im Zustand

$$|\phi\rangle = \frac{1}{\sqrt{2}}|000\rangle + \frac{1}{2}|101\rangle + \frac{1}{2}|111\rangle.$$

Bestimmen Sie für das zweite Qubit die Wahrscheinlichkeiten mit denen $|0\rangle$ bzw. $|1\rangle$ angenommen wird, sowie die Folgezustände.

An welcher Stelle unterscheiden

$$|\phi\rangle = \frac{1}{\sqrt{2}} |1000\rangle + \frac{1}{\sqrt{2}} |101\rangle + \frac{1}{\sqrt{2}} |111\rangle$$

(i) mit Wkeit $(\frac{1}{\sqrt{2}})^2 + (\frac{1}{\sqrt{2}})^2 = \frac{1}{2} + \frac{1}{2} = \frac{3}{4}$ wird $|10\rangle$ angenommen. Das Register ist dann im Zustand

$$\begin{aligned}\underbrace{\frac{1}{\sqrt{2}} |1000\rangle + \frac{1}{\sqrt{2}} |101\rangle}_{\sqrt{\frac{3}{4}}} &= \frac{1}{\sqrt{2}} \cdot \frac{2}{\sqrt{3}} |1000\rangle + \frac{1}{\sqrt{2}} \cdot \frac{2}{\sqrt{3}} |101\rangle = \frac{2}{\sqrt{6}} |1000\rangle + \frac{2}{\sqrt{6}} |101\rangle \\ &= \frac{\sqrt{2}}{\sqrt{3}} |1000\rangle + \frac{1}{\sqrt{3}} |101\rangle = \underline{\frac{\sqrt{2}}{\sqrt{3}} |1000\rangle + \frac{1}{\sqrt{3}} |101\rangle}\end{aligned}$$

(ii) mit Wkeit $(\frac{1}{\sqrt{2}})^2 = \frac{1}{4}$ wird $|1\rangle$ angenommen

$$\frac{1}{\sqrt{2}} |111\rangle = |111\rangle$$

Verschränkung

Im Abschnitt über das Tensorprodukt hatten wir bereits festgestellt, dass sich

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

faktorisieren lässt (bzgl. dem Tensorprodukt), während (wir zumindest behauptete hatten, dass) dies für

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

nicht gilt. Den ersten Zustand hatten wir *separabel* genannt, den zweiten *verschränkt*.

Verschränkung

Wir setzen die zuvor aufgeschobene Diskussion verschränkter Zustände nun fort.

Definition (Verschränkung)

Sei $|\phi\rangle$ der Zustand eines Quantenregisters aus n Bits. Der Zustand $|\phi\rangle$ heißt *unverschränkt* oder *separabel*, wenn er das Produkt von Zuständen einzelner Qubits ist, d.h.

$$|\phi\rangle = |\phi_{n-1}\rangle \otimes |\phi_{n-2}\rangle \otimes \dots \otimes |\phi_1\rangle.$$

Ein Zustand heißt *verschränkt*, wenn es keine solche Zerlegung gibt.

Verschränkung

Messung eines einzelnen Qubits in einem Produktzustand hat keine Auswirkungen auf die übrigen Qubits.

Messen wir das linke Qubit in

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

erhalten wir $|0\rangle$ oder $|1\rangle$ jeweils mit Wahrscheinlichkeit $\frac{1}{2}$.
Folgezustände sind $|0\rangle|-\rangle$ oder $|1\rangle|-\rangle$.

Das rechte Qubit $|-\rangle$ bleibt unverändert und wird durch die Messung des linken Qubits nicht beeinflusst.

Verschränkung

Messung eines einzelnen Qubits in einem verschränkten Zustand kann die anderen Qubits beeinflussen.

Wir betrachten

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

Messen wir das linke Qubit, erhalten wir $|0\rangle$ oder $|1\rangle$ jeweils mit Wahrscheinlichkeit $\frac{1}{2}$. Folgezustände sind $|00\rangle$ oder $|11\rangle$.

Messen wir erneut das linke Qubit (im Folgezustand) und erhalten $|0\rangle$, dann muss auch das rechte Qubit $|0\rangle$ sein.

Analog: Messen wir das linke Qubit (im Folgezustand) und erhalten $|1\rangle$, dann muss auch das rechte Qubit $|1\rangle$ sein.

Verschränkung

Im vorherigen Beispiel hat $|\Phi^+\rangle$ den größtmöglichen Grad an Verschränkung, da die Messung eines einzelnen Qubits die übrigen Qubits vollständig bestimmt. In einem solchen Fall nennt man die Zustände **maximal verschränkt**.

Für zwei Qubits gibt es vier maximal verschränkte Zustände, die sog. **Bell-Zustände**

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

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

Verschränkung

Übung: Zeigen Sie, dass $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ nicht in das Produkt zweier Zustände jeweils eines Qubits zerlegt werden kann.

Angenommen die Zerlegung des Produktes wäre möglich:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \stackrel{!}{=} (\alpha_0|0\rangle + \alpha_1|1\rangle)(\beta_0|0\rangle + \beta_1|1\rangle)$$

Nach Koeffizientenvergleich muss gelten:

$$\alpha_0\beta_0 = \alpha_1\beta_1 = \frac{1}{\sqrt{2}} \quad \text{und} \quad \alpha_0\beta_1 = \alpha_1\beta_0 = 0 \quad \xi$$

⇒ Also ist die Darstellung als Produkt nicht möglich

Verschränkung

Übung: Zeigen Sie, dass $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ nicht in das Produkt zweier Zustände jeweils eines Qubits zerlegt werden kann.

Um $|\Phi^+\rangle$ zu erzeugen, kann man CNOT auf den unverschränkten Zustand $\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$ anwenden und so einen verschränkten Zustand erhalten.

CNOT erzeugt also Verschränkung.

Übung: Zerlegen Sie $\frac{1}{2}(|0\rangle + |3\rangle + |12\rangle + |15\rangle)$ in ein Produkt von Bell-Zuständen.

Verschränkung

Betrachten wir den verschränkten Zustand

$$\frac{\sqrt{3}}{2\sqrt{2}}|00\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|01\rangle + \frac{\sqrt{3}}{4}|10\rangle + \frac{1}{4}|11\rangle$$

und messen das linke Qubit, erhalten wir

- $|0\rangle$ mit Wahrscheinlichkeit $\frac{3}{4}$ und den Folgezustand

$$\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) = |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

- $|1\rangle$ mit Wahrscheinlichkeit $\frac{1}{4}$ und den Folgezustand

$$\frac{\sqrt{3}}{2}|10\rangle + \frac{1}{2}|11\rangle = |1\rangle \otimes \left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle \right)$$

Verschränkung

Messung des linken Qubits hat also das rechte Qubit beeinflusst und in einem Fall erhalten wir $|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ als Folgezustand, im anderen Fall $|1\rangle \otimes \left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle\right)$.

Wird nun das rechte Qubit gemessen, erhalten wir für $|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ jeweils mit Wahrscheinlichkeit $\frac{1}{2}$ die Folgezustände $|0\rangle$ oder $|1\rangle$.

Im Fall von $|1\rangle \otimes \left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle\right)$ erhalten wir $|0\rangle$ mit Wahrscheinlichkeit $\frac{3}{4}$ und $|1\rangle$ mit Wahrscheinlichkeit $\frac{1}{4}$.

Verschränkung

Messung des linken Qubits hat also das rechte Qubit beeinflusst, aber nicht vollständig bestimmt was eine Messung des rechten Qubits ergeben würde.

Der ursprüngliche Zustand ist also nicht maximal verschränkt. Wir nennen einen solchen Zustand *teilweise verschränkt*.

Maximal verschränkte Zustände, wie die Bell-Zustände, sind für unsere Anwendungen interessanter.

Dichte Kodierung

Ausgangssituation: Alice möchte Bob klassische Informationen senden. Genauer: Alice will eine der vier möglichen Nachrichten $N \in \{00, 01, 10, 11\}$ senden. Sie kann dafür einen klassischen Kanal oder einen Quantenkanal verwenden.

Klassischer Kanal: Alice sendet Bob zwei Bits. Weniger als zwei Bits sind nicht möglich, da sie diese benötigt um eine der Kombinationen 00, 01, 10, 11 darzustellen.

Dichte Kodierung

Dichte Kodierung ermöglicht es, mit Hilfe eines Quantenkanals und eines Qubits *zwei* klassische Bits zu übertragen, d.h. es findet eine *Komprimierung* von *zwei* klassischen Bits auf ein Qubit statt.

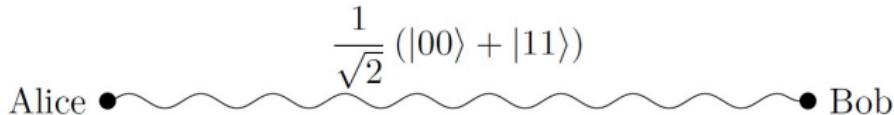
Erkenntnis: Mit einem Qubit lässt sich unter bestimmten Umständen die **doppelte Informationsmenge** wie mit einem **klassischen Bit** übertragen.

Dichte Kodierung

Ausgangssituation: Alice besitzt ein Qubit $|a\rangle$ und Bob ein Qubit $|b\rangle$, die sich in einem verschränkten Zustand

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

befinden.

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


Dichte Kodierung

Kodierung Alice

Alice wendet je nach zu übermittelnder Nachricht eine der folgenden Operationen an:

- $N = 00$, keine Operation auf $|a\rangle$
- $N = 01$, dann $|a\rangle \leftarrow X|a\rangle$
- $N = 10$, dann $|a\rangle \leftarrow Z|a\rangle$
- $N = 11$, dann $|a\rangle \leftarrow ZX|a\rangle$ *//erst X-Gatter, dann Z-Gatter*

Anschließend schickt Alice ihr Qubit $|a\rangle$ an Bob.

Dichte Kodierung

Wir beobachten:

- Für $N = 00$ ist unverändert $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- Für $N = 01$ ist $X|0\rangle = |1\rangle$ und $X|1\rangle = |0\rangle$. Dadurch wird $|\Phi^+\rangle$ zu $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)$.
- Für $N = 10$, ist $Z|0\rangle = |0\rangle$ und $Z|1\rangle = -|1\rangle$. Dadurch wird $|\Phi^+\rangle$ zu $|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$.
- Für $N = 11$, ist $ZX|0\rangle = -|1\rangle$ und $ZX|1\rangle = |0\rangle$. Dadurch wird $|\Phi^+\rangle$ zu $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$.

Dichte Kodierung

Dekodierung Bob

Bob wendet CNOT gefolgt von $H \otimes I$ an, um die Information von Alice zu dekodieren.

Es ist

$$|\Phi^+\rangle \xrightarrow{CNOT} \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) = |+\rangle|0\rangle \xrightarrow{H \otimes I} |00\rangle$$

$$|\Psi^+\rangle \xrightarrow{CNOT} \frac{1}{\sqrt{2}} (|01\rangle + |11\rangle) = |+\rangle|1\rangle \xrightarrow{H \otimes I} |01\rangle$$

$$|\Phi^-\rangle \xrightarrow{CNOT} \frac{1}{\sqrt{2}} (|00\rangle - |10\rangle) = |-\rangle|0\rangle \xrightarrow{H \otimes I} |10\rangle$$

$$|\Psi^-\rangle \xrightarrow{CNOT} \frac{1}{\sqrt{2}} (|01\rangle - |11\rangle) = |-\rangle|1\rangle \xrightarrow{H \otimes I} |11\rangle$$

Dichte Kodierung

Auf maximal verschränkte Zustände setzt auch das Verfahren der *Quantenteleportation*⁴.

Während bei der *dichten Kodierung* klassische Information mit Hilfe eines Quantenkanals übertragen wurde, wird bei der Quantenteleportation Quanteninformation mit Hilfe eines klassischen Kanals übertragen.

Zurück zu Quantenalgorithmen: Wir wollen als nächstes die Grundidee vom Algorithmus von Deutsch auf größere Register von Qubits übertragen und benötigen dafür die Hadamard Transformation auf Registern.

⁴ vgl. Übungsaufgabe

Algorithmus von Deutsch-Jozsa

Aus der Analyse des n -Bit Zufallsgenerators ist

$$H_n |0\dots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$$

bekannt. Das ist offensichtlich der Spezialfall für das Register $R = |0\dots0\rangle$ der Länge n .

Ein allgemeineres Verständnis von H_n ermöglicht uns die Diskussion fortgeschrittenerer Algorithmen bzw. Schaltkreise.

Algorithmus von Deutsch-Jozsa

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

Wir untersuchen die Wirkung von H_n auf ein Register

$R = |x_{n-1}...x_0\rangle$ mit $x_i \in \{0, 1\}$ für $i = 0, \dots, n - 1$ und beginnen mit dem Fall $n = 2$:

$$\begin{aligned}H_2|x_1x_0\rangle &= (H \otimes H)|x_1x_0\rangle \\&= \frac{1}{2}(|0\rangle + (-1)^{x_1}|1\rangle)(|0\rangle + (-1)^{x_0}|1\rangle)\end{aligned}$$

und mit $(-1)^{x_0}(-1)^{x_1} = (-1)^{x_0 \oplus x_1}$ folgt für $x = (x_1, x_0)^T$

Algorithmus von Deutsch-Jozsa

$$\begin{aligned} H_2|x_1x_0\rangle &= \frac{1}{2} (|00\rangle + (-1)^{x_0}|01\rangle + (-1)^{x_1}|10\rangle + (-1)^{x_0 \oplus x_1}|11\rangle) \\ &= \frac{1}{2}((-1)^{(0,0)\cdot\mathbf{x}}|00\rangle + (-1)^{(0,1)\cdot\mathbf{x}}|01\rangle + (-1)^{(1,0)\cdot\mathbf{x}}|10\rangle + \\ &\quad + (-1)^{(1,1)\cdot\mathbf{x}}|11\rangle), \end{aligned}$$

wenn \cdot das Skalarprodukt zweier Vektoren bezeichnet.

Allgemein ist für zwei Vektoren $x, y \in \{0, 1\}^n$ das Skalarprodukt $x \cdot y$ durch $\bigoplus_{i=1}^n x_i y_i$ gegeben. Auf ein n -Bit Register im Zustand $\mathbf{x} \in \{0, 1\}^n$ hat die Hadamard-Transformation H_n die Wirkung

$$H_n|\mathbf{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{y} \in \{0,1\}^n} (-1)^{\mathbf{x} \cdot \mathbf{y}} |\mathbf{y}\rangle.$$

Algorithmus von Deutsch-Jozsa

Dies ist eine Superposition über alle klassischen Zustände des Registers. Die Information über $|x\rangle$ wird in die Amplitude verlagert.

Beispiel: Als *gleichgewichtete* Superposition bezeichnet man

$$H_n|0\dots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^0 |y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} |y\rangle.$$

Die *alternierende* Superposition ist für $y = (y_{n-1}, \dots, y_0)$

$$H_n|0\dots01\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{y_0} |y\rangle,$$

Algorithmus von Deutsch-Jozsa

wobei für die Summe gilt

$$\begin{aligned} \sum_{y=0}^{2^n-1} (-1)^{y0}|y\rangle &= (-1)^0|0\dots00\rangle + (-1)^1|0\dots01\rangle + \\ &\quad + (-1)^0|0\dots010\rangle + (-1)^1|0\dots011\rangle + \dots \\ &\quad + (-1)^0|1\dots10\rangle + (-1)^1|1\dots11\rangle \\ &= |0\dots00\rangle - |0\dots01\rangle + |0\dots010\rangle - |0\dots011\rangle + \dots \\ &\quad + |1\dots10\rangle - |1\dots11\rangle. \end{aligned}$$

Die allgemeine Formel für die Hadamard Transformation werden wir noch häufig benötigen.

Algorithmus von Deutsch-Jozsa

Wollen wir Quantenalgorithmen besser verstehen und mögliche Vorteile identifizieren benötigen wir Vergleichsgrößen.

Für Schaltkreise wäre es am präzisesten, die kleinstmögliche Anzahl von Gattern (bzgl. einer Gatterbasis) zu zählen, die zum Lösen einer Problemstellung benötigt werden. Diese Größe nennt man *Schaltkreiskomplexität*.

In diesem Sinne wäre ein Quantenalgorithmus *effizient*, wenn er **polynomiale Schaltkreiskomplexität** hätte.

Die Schaltkreiskomplexität zu bestimmen ist mitunter schwer.

Algorithmus von Deutsch-Jozsa

Wir verwenden die *Anfragekomplexität* (*query complexity*).

Das ist die Anzahl der Aufrufe bzw. Anfragen an eine Funktion bzw. ein Orakel, die benötigt wird um ein Problem zu lösen.

Das Orakel ist für uns eine Black Box, an die wir einen Input übergeben und die uns eine Output zurück liefert, ohne das wir näher wissen was im Inneren des Orakel passiert. Wir können dies als eine externe (Quanten-)Subroutine verstehen.

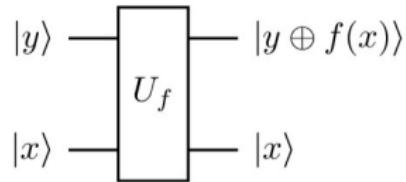
Algorithmus von Deutsch-Jozsa

Tatsächlich kennen wir diese Arbeitsweise schon vom Algorithmus von Deutsch. Dort wurde U_f verwendet, für $f : \{0, 1\} \rightarrow \{0, 1\}$.

Ein Quantenorakel ist

$$U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle.$$

Das Qubit $|y\rangle$ nennt man
Antwortqubit oder *target Qubit*;
 $|x\rangle$ nennt man *input Qubit*.



Algorithmus von Deutsch-Jozsa

Bei zuvor beschriebener Anfrage an ein Orakel wird das input Qubit $|x\rangle$ nicht verändert, während das target Qubit von $|y\rangle$ zu $|y \oplus f(x)\rangle$ übergeht.

Im Algorithmus von Deutsch haben wir (indirekt) schon eine Möglichkeit kennengelernt um Anfragen zu stellen, bei denen das target Qubit unverändert bleibt, während das input Qubit $|x\rangle$ mit einer Phase multipliziert wird:

Initialisiere das target Qubit mit $|0\rangle$ und

$$|x\rangle|0\rangle \xrightarrow{I \otimes X} |x\rangle|1\rangle \xrightarrow{I \otimes H} |x\rangle|-\rangle.$$

Algorithmus von Deutsch-Jozsa

Dies führt weiter zu

$$\begin{aligned}|x\rangle|-\rangle &= |x\rangle \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}} (|x\rangle|0\rangle - |x\rangle|1\rangle) \\&\xrightarrow{U_f} \frac{1}{\sqrt{2}} (|x\rangle|f(x)\rangle - |x\rangle|1 \oplus f(x)\rangle) \\&= \begin{cases} \frac{1}{\sqrt{2}} (|x\rangle|0\rangle - |x\rangle|1\rangle) = |x\rangle|-\rangle & \text{falls } f(x) = 0 \\ \frac{1}{\sqrt{2}} (|x\rangle|1\rangle - |x\rangle|0\rangle) = -|x\rangle|-\rangle & \text{falls } f(x) = 1 \end{cases} \\&= (-1)^{f(x)}|x\rangle|-\rangle\end{aligned}$$

Da die Funktionsauswertung des input Qubit nun als Phase auftritt, nennt man dies auch **phase kickback**.

Algorithmus von Deutsch-Jozsa

Manchmal wird auch die Bezeichnung **Phasenorakel** verwendet.

Das ist insbesondere dann der Fall, wenn die Notation von $|-\rangle$ vernachlässigt wird und man abkürzend

$$|x\rangle \xrightarrow{U_f} (-1)^{f(x)}|x\rangle$$

schreibt.

Algorithmus von Deutsch-Jozsa

Ausgangssituation: Gegeben sei eine Funktion

$f : \{0,1\}^n \rightarrow \{0,1\}$ für die genau eine der folgenden beiden Möglichkeiten gilt:

- f ist *konstant*, d.h. alle Eingaben werden auf die gleiche Ausgabe abgebildet, oder
- f ist *balanciert*, d.h. die Hälfte der Eingaben wird auf 1 abgebildet und die andere Hälfte wird auf 0 abgebildet. Also $|f^{-1}(0)| = |f^{-1}(1)| = 2^{n-1}$ für die Urbilder.

Das Quantenorakel ist

$\downarrow i^n$ } ist es 0
 i } ist es 1

$$U_f : |x_{n-1} \dots x_0, y\rangle \mapsto |x_{n-1} \dots x_0, y \oplus f(x)\rangle.$$

Frage: Ist f konstant oder balanciert?

Algorithmus von Deutsch-Jozsa

Klassische Lösung: Um mit Sicherheit zu bestimmen welche der Möglichkeiten der Fall ist, müssen wir die Hälfte der Inputs abfragen, plus einen zusätzlichen Input. Wurde für die Hälfte der Inputs eine 0 als Ausgabe erhalten, dann löst die zusätzliche Abfrage eines weiteren Inputs die Frage, ob alle Outputs 0 sind oder nicht.

Für einen binären string der Länge n gibt es 2^n mögliche Belegungen, d.h. die Anfragekomplexität an f ist

$$2^{n-1} + 1 = O(2^n)$$

also exponentiell.

Algorithmus von Deutsch-Jozsa

Es gibt probabilistische Algorithmen (vgl. BPP), die die richtige Antwort mit einem beschränkten Fehler in einer konstanten Anzahl der Anfragen an f liefern. In diesem Fall ist die Komplexität $O(1)$.

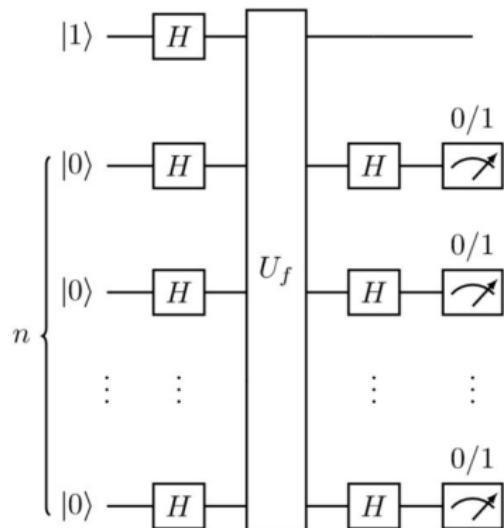
Quantenalgorithmus: Der Algorithmus von Deutsch-Jozsa löst das Problem mit *genau einer* Anfrage.

Das ist eine exponentielle Beschleunigung gegenüber dem exakten klassischen Lösungsansatz, aber nicht gegenüber einem probabilistischen klassischen Algorithmus.

Algorithmus von Deutsch-Jozsa

Algorithmus von Deutsch-Jozsa

1. $|x_{n-1} \dots x_0\rangle |y\rangle \leftarrow |0 \dots 0\rangle |1\rangle$
2. $|x\rangle |y\rangle \leftarrow H_{n+1} |x\rangle |y\rangle$
3. $|x\rangle |y\rangle \leftarrow U_f |x\rangle |y\rangle$
4. $|x\rangle |y\rangle \leftarrow (H_n |x\rangle) |y\rangle$
5. Messe das Register $|x\rangle$:
 - Ist $|x\rangle = |0 \dots 0\rangle$: f ist konstant
 - Sonst: f ist balanciert



Algorithmus von Deutsch-Jozsa

Analyse des Algorithmus: In Schritt 2 ergibt sich

$$|\phi_2\rangle = H_n|0\dots0\rangle H|1\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \right) |- \rangle.$$

Wir erkennen die gleichgewichtete Superposition. Mittels phase kickback ist

$$U_f(|x\rangle |- \rangle) = (-1)^{f(x)} |x\rangle |- \rangle.$$

Algorithmus von Deutsch-Jozsa

Damit folgt in **Schritt 3**

$$\begin{aligned} |\phi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} U_f(|x\rangle |-\rangle) \\ &= \left(\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle \right) |-\rangle. \end{aligned}$$

In **Schritt 4** erfolgt erneute Anwendung von H_n (diesmal nur auf n Register):

$$|\phi_4\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} H_n |x\rangle \right) |-\rangle.$$

Algorithmus von Deutsch-Jozsa

Entsprechend der Vorbetrachtung zur Hadamard- Transformation auf **allgemeinen** Registern ist

$$\begin{aligned} |\phi_4\rangle &= \left(\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} \left(\frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle \right) \right) |- \rangle \\ &= \left(\sum_{z \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)+x \cdot z} \right) |z\rangle \right) |- \rangle. \end{aligned}$$

Um zu verstehen wie eine Messung uns bestimmen lässt, ob die Funktion konstant oder balanciert ist, berechnen wir die Wahrscheinlichkeit den Zustand $|z\rangle = |0\dots00\rangle$ zu messen.

Algorithmus von Deutsch-Jozsa

Zu beachten ist, dass nicht alle Qubits gemessen werden (vgl. Beschreibung des Algorithmus).

Die Amplitude von $|0\dots00\rangle$ vor der Messung ist

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)+x \cdot 0} = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)}.$$

Dies bestimmt schon, ob f konstant oder balanciert ist.

Algorithmus von Deutsch-Jozsa

1. Fall: Ist f konstant, dann nimmt $f(x)$ immer den gleichen Wert an, also $f(x) = f(0\dots 00)$ für alle $x \in \{0,1\}^n$. In diesem Fall ist

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(0\dots 00)} = (-1)^{f(0\dots 00)}.$$

Ist f konstant, dann ist die Wahrscheinlichkeit $|0\dots 00\rangle$ zu messen, also 1.

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(0\dots 00)} = \frac{(-1)^{f(0\dots 00)}}{2^n} \cdot \sum_{x \in \{0,1\}^n} 1 = \frac{(-1)^{f(0\dots 00)}}{2^n} \cdot 2^n = (-1)^{f(0\dots 00)}$$

Algorithmus von Deutsch-Jozsa

2. Fall: Ist f **balanciert**, dann ist $(-1)^{f(x)}$ in der Hälfte der Fälle 1 und in der anderen Hälfte der Fälle -1 . In diesem Fall ist die Summe alternierend und

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} = 0.$$

Ist f balanciert, dann ist die Wahrscheinlichkeit $|0\dots00\rangle$ zu messen demnach 0. Wir erhalten bei Messung also garantiert etwas anderes als $|0\dots00\rangle$.

Algorithmus von Deutsch-Jozsa

Um zu bestimmen ob f konstant oder balanciert ist, genügt es also n Qubits zu messen.

Erhalten wir dabei $|0\dots00\rangle$, dann ist f konstant. Ergibt unsere Messung irgendeinen anderes Ergebnis, dann ist f balanciert.

Um die Ausgangsfrage zu beantworten, war genau eine Anfrage an das Quantenorakel notwendig.

Algorithmus von Bernstein-Vazirani

Ein weiterer Fragestellung die sich mit derselben Prozedur wie im Deutsch-Jozsa Algorithmus lösen lässt, ist die Frage nach der Bestimmung eines unbekannten n -Bit Strings, der als Punktprodukt gegeben ist.

Gegeben $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Wir wissen das $f(x) = s \cdot x$, wobei $s = s_{n-1} \dots s_1 s_0$ ein unbekannter String ist und (wie üblich)

$$s \cdot x = s_{n-1}x_{n-1} \oplus \dots \oplus s_1x_1 \oplus s_0x_0.$$

Problemstellung: Bestimme $s = s_{n-1} \dots s_1 s_0$.

Algorithmus von Bernstein-Vazirani

Klassische Lösung: Es werden n Anfragen an f benötigt um jedes Bit von s zu lernen.

Für $n = 3$ sind dies die Anfragen

$$f(001) = s_2 \cdot 0 + s_1 \cdot 0 + s_0 \cdot 1 = s_0$$

$$f(010) = s_2 \cdot 0 + s_1 \cdot 1 + s_0 \cdot 0 = s_1$$

$$f(100) = s_2 \cdot 1 + s_1 \cdot 0 + s_0 \cdot 0 = s_2$$

Auch klassische probabilistische Algorithmen benötigen für dieses Problem mindestens n Anfragen an f .

Algorithmus von Bernstein-Vazirani

Quantenalgorithmus: Der Algorithmus von Bernstein-Vazirani benötigt *genau eine* Anfrage. Der Algorithmus bzw. Schaltkreis ist exakt der gleiche wie im Algorithmus von Deutsch-Jozsa, nur das nun eine spezielle Form von f vorgegeben ist.

Die Analyse des Algorithmus ist eine Aufgabe auf dem Übungsblatt.

Da klassische Algorithmen n Anfragen benötigen, der Algorithmus von Bernstein-Vazirani jedoch nur eine einzige Anfrage, liefert dieser eine *polynomielle* Beschleunigung gegenüber klassischen Computern.