Quantencomputing

Modul 7270

Martin Rehberg

Hessen3C / Hochschule RheinMain

Inhaltsverzeichnis

Inhaltsverzeichnis

Einleitung

Ziele von Quantencomputing:

- Quantencomputer bauen
- Quantenalgorithmen entwickeln / untersuchen

Einleitung

Ziele von Quantencomputing:

- Quantencomputer bauen
- Quantenalgorithmen entwickeln / untersuchen

Ziel der Vorlesung:

- Einführung in die grundlegende Funktionsweise von Quantencomputern
  - physikalischen Grundlagen als *gegeben* annehmen
  - Mathematik werden wir nach Bedarf erarbeiten / wiederholen
- Anwendungen mit Blick auf Verschlüsselungsverfahren

Einleitung

Literatur:[1]

- Matthias Homeister - Quantencomputing verstehen (**Hauptquelle**), 5. Auflage, Springer, 2018.
- Artuhr Pittenger - An Indroduction to Quantum Computing Algorithms, Birkhäuser, 2001.
- Michael Nielsen, Isaac Chuang - Quantum Computation and Quantum Information, 10. Auflage, Cambridge University Press, 2010.
- Dirk Hoffmann - Theoretische Informatik, 2. Auflage, Hanser, 2011.

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

Einleitung

Klassische Welt

- mechanische Rechenmaschinen
  - Difference Engine, Analytical Engine - Charles Babbage
  - Schachmaschine - Leonardo Quevedo
- elektromechanische Rechenmaschinen
  - Z3, Z4 - Konrad Zuse
  - Kryptoanalyse - 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
- 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

Verschränkung von Quantenbits hat weitreichende Folgen, etwa

- Primfaktorisierung $\rightsquigarrow$ RSA-Verfahren
- diskreter Logarithmus $\rightsquigarrow$ Elliptic Curve Diffie-Hellman
- Suche in Datenbanken, u.v.m.

Einleitung

Verschränkung von Quantenbits hat weitreichende Folgen, etwa

- Primfaktorisierung $\rightsquigarrow$ RSA-Verfahren
- diskreter Logarithmus $\rightsquigarrow$ Elliptic Curve Diffie-Hellman
- Suche in Datenbanken, u.v.m.

Es gibt aber nicht nur Vorteile:

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

Berechenbarkeit und Turingmaschinen

Berechnung (intuitiv)

Einem *Berechnungsgerät* wir eine Eingabe übergeben. Anschließend führt das Gerät deterministisch Berechnungen durch.
Eine *Berechnung* ist eine Folge von Zuständen des Berechnungsgerätes. Jeder Rechenschritt ist ein Übergang zwischen den Zuständen und hängt allein vom aktuellen Zustand ab.

Berechenbarkeit und Turingmaschinen

Definition: Alphabet, Zeichen, Wort, formale Sprache

- Ein *Alphabet* Σ ist eine endliche Menge von Symbolen.
- Ein Element $\sigma \in \Sigma$ heißt *Zeichen* des Alphabets.
- Ein Element $\omega \in \Sigma^* := \bigcup_{i=0}^{\infty} \Sigma^i$ wird *Wort* über Σ genannt, wobei $\Sigma^0 := \{\varepsilon\}$. Man nennt ε das *leere Wort*.
- Eine Teilmenge $L \subseteq \Sigma^*$ wird *formale Sprache* über Σ genannt.

Berechenbarkeit und Turingmaschinen

Definition: Alphabet, Zeichen, Wort, formale Sprache

- Ein *Alphabet* Σ ist eine endliche Menge von Symbolen.
- Ein Element $\sigma \in \Sigma$ heißt *Zeichen* des Alphabets.
- Ein Element $\omega \in \Sigma^* := \bigcup_{i=0}^{\infty} \Sigma^i$ wird *Wort* über Σ genannt, wobei $\Sigma^0 := \{\varepsilon\}$. Man nennt ε das *leere Wort*.
- Eine Teilmenge $L \subseteq \Sigma^*$ wird *formale Sprache* über Σ genannt.

Beispiel Palindromsprache

Für $\Sigma = \{a, b, c, ..., x, y, z\}$ sei L die Menge aller spiegelbildlich angeordneten Zeichenketten. In dieser Sprache sind die Wörter *aabaa*, *anna* und *otto* enthalten. Nicht enthalten sind *abab*, *abc* oder *aaba*.

Berechenbarkeit und Turingmaschinen

Definition: Turingmaschine

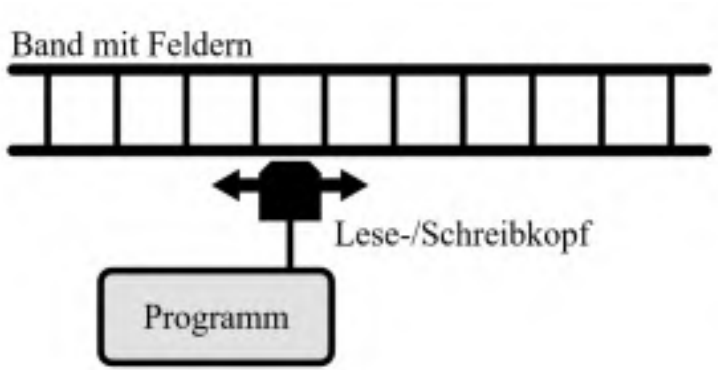
Eine (deterministische) *Turingmaschine* (TM) ist ein 7-Tupel $(S, \Sigma, \Pi, \delta, s_0, \square, E)$, bestehend aus

- der endlichen *Zustandsmenge* S,
- dem endlichen *Eingabealphabet* Σ,
- dem *Bandalphabet* Π mit $\Pi \supset \Sigma$,
- der *Zustandsübergangsfunktion* $\delta : S \times \Pi \to S \times \Pi \times \{\leftarrow, \rightarrow\}$,
- dem *Startzustand* s_0,
- dem *Blank-Symbol* $\square \in \Pi \setminus \Sigma$,
- der Menge der *Endzustände* $E \subseteq S$.

Berechenbarkeit und Turingmaschinen

Startkonfiguration:

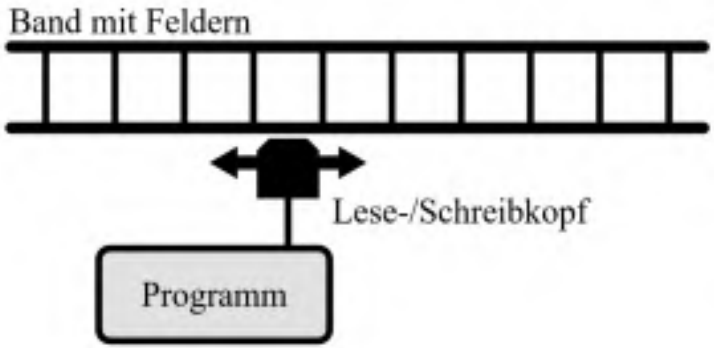
- TM ist im Startzustand s_0
- zu verarbeitendes Eingabewort $\omega \in \Sigma^*$ steht auf dem Band
- Lese-/Schreibkopf über dem ersten Eingabezeichen positioniert
- alle Felder links und rechts des Eingabewortes sind mit dem Blank-Symbol $\square$ beschrieben

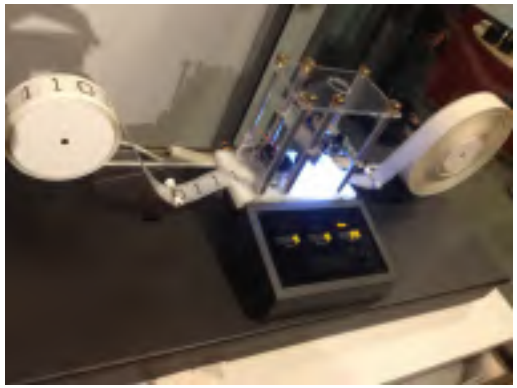


Berechenbarkeit und Turingmaschinen

Programmablauf:

- Der Lese-/Schreibkopf liest das aktuelle Bandzeichen σ ein
- Der Funktionswert $(s', \sigma', r) = \delta(s, \sigma)$ wird berechnet
- Das Bandzeichen wird durch σ' ersetzt
- Der Kopf wird nach links ($\leftarrow$) oder rechts ($\rightarrow$) bewegt
- Der Folgezustand s' wird angenommen





Berechenbarkeit und Turingmaschinen

Definition: (Turing-) Berechenbarkeit

Eine Funktion $f : \Sigma^* \to \Sigma^*$ heißt *(turing-) berechenbar*, wenn eine TM $T = (S, \Sigma, \Pi, \delta, s_0, \square, E)$ existiert, die für alle $\omega \in \Sigma^*$ mit $f(\omega)$ auf dem Band anhält oder in eine Endlosschleife gerät, wenn $f(\omega)$ nicht definiert ist.

Variationen von TM:

- mehrere Bänder / Folgezustände
- Folgezustände per Münzwurf

Alan Turing

Berechenbarkeit und Turingmaschinen

These von Church (1936)

Jede im intuitiven Sinn berechenbare Funktion ist durch eine Turingmaschine berechenbar.

Mehr noch: Alles was mit einem QC berechenbar ist, kann auch mit einer TM berechnet werden.
Aber: Wahrscheinlich gibt es praktisch relevante Probleme, die mit QC *schneller* gelöst werden können.

Alonzo Church

Grundlagen der Quantenmechanik

Ziel: Eine Idee für die Beobachtungen der Physik gewinnen, nicht aber die physikalischen Beobachtungen in der Quantenwelt erklären.

Wir wollen die Begriffe *Superposition* und *Messen* veranschaulichen.

Gedankenexperiment:
Schrödingers Katze

Erwin Schrödinger

Grundlagen der Quantenmechanik

Klassisch: Eine Katze sitzt in einer undurchsichtigen Box. Diese enthält einen (klassischen) Mechanismus, der die Katze mit Wahrscheinlichkeit $1/2$ sofort tötet.

Die Katze ist *entweder* tot *oder* lebendig.

Grundlagen der Quantenmechanik

Modifikation: Der Mechanismus wird mit einem quantenmechanischen Prozess gekoppelt, etwa dem Zerfall eines radioaktiven Atoms.

Quantenmechanisch: Das Atom ist *gleichzeitig* unverändert bzw. zerfallen, also ist die Katze *gleichzeitig* tot und lebendig (Zustand der Superposition). Wird die Box geöffnet, dann ist die Katze *entweder* tot *oder* lebendig. Das öffnen der Box entspricht dem Messen.

Grundlagen der Quantenmechanik

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

Grundlagen der Quantenmechanik

Beispiel: Zulässige Zustände sind $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ oder $\frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$, denn $\left(\frac{1}{\sqrt{2}}\right)^2 + \left(\frac{1}{\sqrt{2}}\right)^2 = 1$ bzw. $\left(\frac{1}{\sqrt{3}}\right)^2 + \left(\sqrt{\frac{2}{3}}\right)^2 = 1$.

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.

Grundlagen der Quantenmechanik

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.

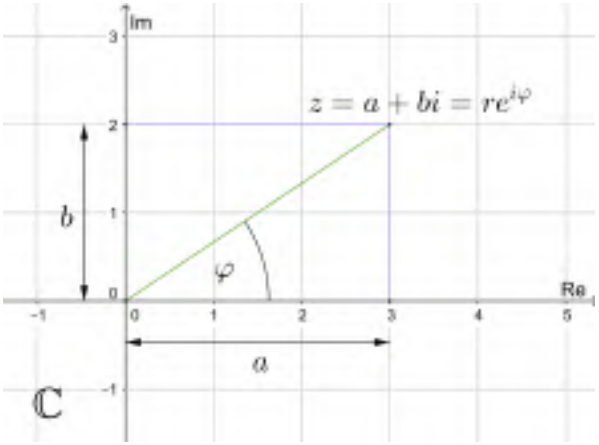
Beispiel: Das Qubit $\frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$ ist nach dem Messen mit Wahrscheinlichkeit $1/3$ im Zustand $|0\rangle$ und mit Wahrscheinlichkeit $2/3$ im Zustand $|1\rangle$.

Übung: Was beobachten Sie beim Messen der Qubits $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ und $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$?

Grundlagen der Quantenmechanik

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 $\overline{z} := a - ib$ heißt die *Konjugierte* von z. Der *Betrag* einer komplexen Zahl ist $|z| := \sqrt{a^2 + b^2}$.

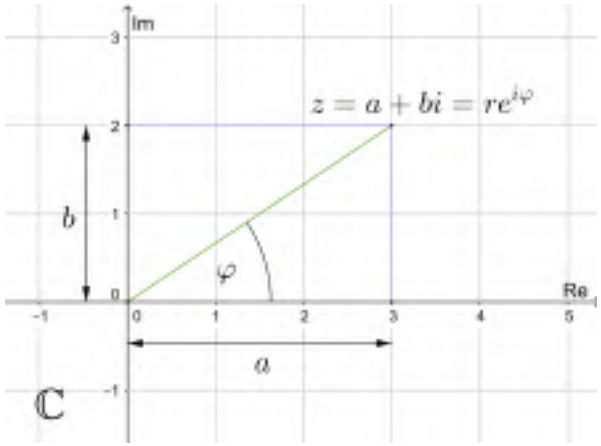
Die *Polarkoordinatendarstellung* einer komplexen Zahl ist $z = re^{i\varphi}$, wobei r der Betrag ist und φ die *Phase*.



Grundlagen der Quantenmechanik

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 $\mathbf{1} = (1, 0)$ und $i = (0, 1)$.



Grundlagen der Quantenmechanik

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 $\mathbb{C}^2$.

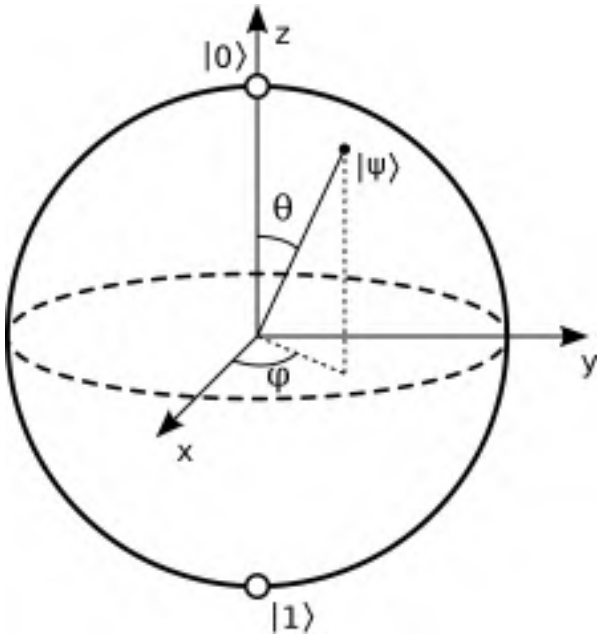
Grundlagen der Quantenmechanik

Mittels

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

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

Das Bild erinnert uns an die komplexen Zahlen mit der *Riemannschen Zahlenkugel*.



Grundlagen der Quantenmechanik

Rechenschritte auf Qubits: *unitäre* Matrizen (physikalisch begr.)

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.

Grundlagen der Quantenmechanik

Definition (konjugierte und adjungierte Matrix)

Sei $A = (a_{ij}) \in \mathbb{C}^{m \times n}$. Die Matrix $\overline{A} := (\overline{a_{ij}}) \in \mathbb{C}^{m \times n}$ heißt die *Konjugierte* von A, und $A^\dagger := (\overline{A})^T$ 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 sofort das unitäre Matrizen *invertierbar* sind, denn nach Definition gilt $A^\dagger A = A A^\dagger = I_n$.

Grundlagen der Quantenmechanik

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

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

Grundlagen der Quantenmechanik

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

Grundlagen der Quantenmechanik

Wir untersuchen die Wirkung der Hadamard-Transformation auf den Basiszuständen $|0\rangle$ und $|1\rangle$. 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

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

Analog:

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

Grundlagen der Quantenmechanik

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

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

und

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

Übung: Konstruieren Sie alle unitäten Transformationen A, für die gilt

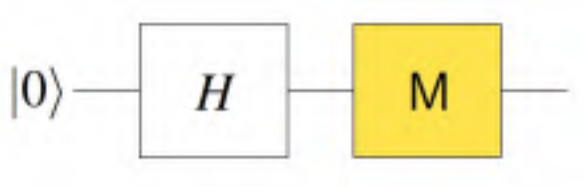
$$|0\rangle \xrightarrow{A} \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle.$$

Quantenzufallsgenerator und das Problem von Deutsch

Erste Anwendung: Ein (echter) Zufallsgenerator

Algorithmus: Zufallsgenerator

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 $|1\rangle$.

Quantenzufallsgenerator und das Problem von Deutsch

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?

Quantenzufallsgenerator und das Problem von Deutsch

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?

Eine Hadwareumsetzung eines solchen Zufallsgenerators vertreibt etwa die Schweizer Firma ID Quantique. Dabei sendet eine Einzelphotonenquelle Lichtteilchen aus, die auf einen halbtransparenten Spiegel treffen. Mit jeweils Wahrscheinlichkeit $1/2$ passiert das Photon dieses Bauteil oder wird reflektiert.

Quantenzufallsgenerator und das Problem von Deutsch

Ein Bit ist für komplexere Anwendungen nicht ausreichend
$\Rightarrow$ Quantenregister

Der Inhalt eines 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.

Quantenzufallsgenerator und das Problem von Deutsch

Beispiel: 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:

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

Quantenzufallsgenerator und das Problem von Deutsch

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: Zeige: 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 $\alpha_0, \alpha_1, \alpha_2, \alpha_3$.

Quantenzufallsgenerator und das Problem von Deutsch

Definition (Quantenregister)

Ein *Quantenregister* R der Länge $n \geq 1$ hat die Form $R = |x_{n-1}\rangle|x_{n-2}\rangle...|x_0\rangle = |x_{n-1}x_{n-2}...x_0\rangle$. Es kann sich in einem Zustand der Form $\sum_{i=0}^{2^n-1} \alpha_i|i\rangle$ befinden, wobei $|i\rangle = |\,\mathrm{bin}(i)\rangle$ und $\alpha_i \in \mathbb{C}$ für $i = 0, ..., 2^n - 1$ 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 $|\alpha_i|^2$.

Quantenzufallsgenerator und das Problem von Deutsch

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

$$|0...00\rangle, |0...01\rangle, ..., |1...11\rangle$$

bzw. $|0\rangle, |1\rangle, ..., |2^n - 1\rangle$.

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

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

Quantenzufallsgenerator und das Problem von Deutsch

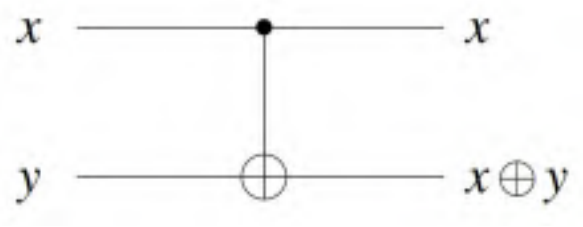
Da die Zustandsvektoren eines n-Bit Quantenregisters je 2^n Komponenten besitzen, entsprechen die einzelnen Rechenschritte einer Quantenmaschine unitären Transformationen, die durch unitäre $2^n \times 2^n$-Matrizen darstellbar sind.

Beispiel: Für $n = 2$ betrachte die *controlled not* Operation

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

Matrixdarstellung:

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



Quantenzufallsgenerator und das Problem von Deutsch

Übung:

1. Zeigen Sie, dass A_{CNOT} unitär ist.
2. Sei P eine quadratische Matrix die in jeder Zeile und jeder Spalte genau einen Eintrag 1 und sonst nur Nullen enthält (solche Matrizen heißen *Permutationsmatrizen*). Zeigen Sie, dass P unitär ist.

Quantenzufallsgenerator und das Problem von Deutsch

Unitäre Transformationen (Matrizen) sind für uns von wesentlicher Bedeutung.

Erinnerung: Eine Matrix A ist unitär, wenn $A^{-1} = A^{\dagger}$ gilt.

Definition (Skalarprodukt, Norm)

Seien $|u\rangle = (\alpha_0, ..., \alpha_{n-1})^T$ und $|v\rangle = (\beta_0, ..., \beta_{n-1})^T$ komplexe Vektoren mit n Komponenten. Das *Skalarprodukt* $\langle u, v\rangle$ (*Braket*-Notation) ist definiert durch

$$\langle u|v\rangle := \overline{\alpha_0}\beta_0 + ... + \overline{\alpha_{n-1}}\beta_{n-1}.$$

Die *Norm* des Vektors $|u\rangle$ ist $\||u\rangle\| := \sqrt{\langle u|u\rangle}$.

Quantenzufallsgenerator und das Problem von Deutsch

Eigenschaften unitärer Transformationen

Sei U eine unitäre Transformation und $|\varphi\rangle, |\psi\rangle$ zwei Vektoren.

1. Unitäre Transformationen sind längenerhaltend, d.h.

$$\|U|\varphi\rangle\| = \||\varphi\rangle\|.$$

2. Unitäre Transformationen ändern das Skalarprodukt nicht, d.h.

$$\langle U\varphi|U\psi\rangle = \langle\varphi|\psi\rangle.$$

3. Unitäre Transformationen sind umkehrbar, d.h. jeder Schritt in einer Berechnung durch einen Quantencomputer kann rückgängig gemacht werden.

Quantenzufallsgenerator und das Problem von Deutsch

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

Ziel: Eine echte Münze (Kopf und Zahl) von einer gefälschten Münze (beide Seiten Kopf) unterscheiden.

Klassisch: Die Münze muss zweimal betrachtet werden, je einmal von jeder Seite.

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

Quantenzufallsgenerator und das Problem von Deutsch

Abstrakt: Gegeben sei eine Funktion $f : \{0,1\} \to \{0,1\}$ 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.
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.

Quantenzufallsgenerator und das Problem von Deutsch

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

Wir verwenden

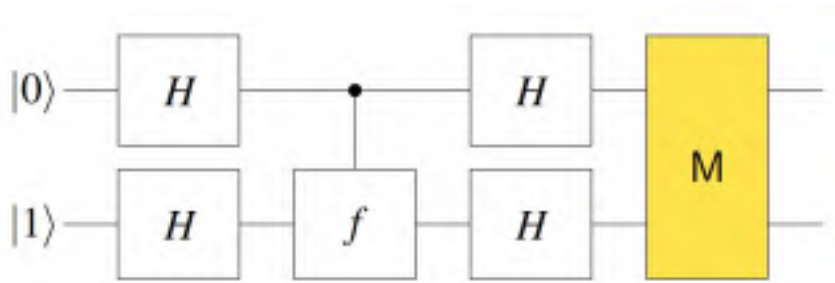
$$U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle.$$

Übung: Zeige, das U_f unitär ist.

Quantenzufallsgenerator und das Problem von Deutsch

Algorithmus: Problem 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. $|x\rangle|y\rangle \leftarrow \mathrm{U_f}\,|x\rangle|y\rangle$
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



Quantenzufallsgenerator und das Problem von Deutsch

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

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

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

Quantenzufallsgenerator und das Problem von Deutsch

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

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

Beobachtung: Bei einer Messung zum jetzigen Zeitpunkt ist jeder der Werte $|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$ mit einer Wahrscheinlichkeit von $1/4$ gleichwahrscheinlich.

Quantenzufallsgenerator und das Problem von Deutsch

Wir können die Terme weiter vereinfachen...

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

... und erhalten

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

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

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

Quantenzufallsgenerator und das Problem 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 $f(0) = f(1) = 1$.

Quantenzufallsgenerator und das Problem 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 $f(0) = 0, f(1) = 1$;

Quantenzufallsgenerator und das Problem 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.

Tensorprodukt, Messen von Registern & Verschränkung

Bisher haben wir mehrere Bits (naiv) zu einem Register zusammengefügt. Um auch mehrere Rechenschritte zu *Operationen auf einem Register* zusammenzufügen, müssen wir unser bisheriges Sichtweise präzisieren.

Vorgehen: Wir überlegen uns wie wir Register aus einzelnen Bits (formal) zusammensetzen. Damit wollen wir Operationen auf einem Register durch Operationen auf einzelnen Bits zusammensetzen.

Erinnerung: Der Zustand eines Quantenregisters aus n Bits wird durch einen 2^n-dimesionalen Vektor beschrieben.
Wir wissen auch: Bits sind Linearkombinationen von Basisvektoren.

Tensorprodukt, Messen von Registern & Verschränkung

Definition (Tensorprodukt von Vektorräumen - Teil 1)

Sei V_1 ein $\mathbb{C}$-Vektorraum mit Basis $\{e_0, ..., e_{m-1}\}$ und V_2 ein $\mathbb{C}$-Vektorraum mit Basis $\{f_0, ..., f_{n-1}\}$. 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, Messen von Registern & Verschränkung

Definition (Tensorprodukt von Vektorräumen - Teil 2)

Ist

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

und

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

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, Messen von Registern & Verschränkung

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, Messen von Registern & Verschränkung

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

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

Tensorprodukt, Messen von Registern & Verschränkung

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

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

Uns interessieren unitäre Transformationen, also unitäre Matrizen.

Tensorprodukt, Messen von Registern & Verschränkung

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, Messen von Registern & Verschränkung

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 ergibt sich

Definition (H_n)

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

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

Tensorprodukt, Messen von Registern & Verschränkung

Übung: Berechnen Sie H_2.

Wir beobachten, dass die folgenden Aktionen auf dasselbe Resultat führen

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

Damit haben wir (wie gewünscht) die Möglichkeit, Operationen auf Registern (statt auf einzelnen Bits) auszuführen.

Tensorprodukt, Messen von Registern & Verschränkung

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?

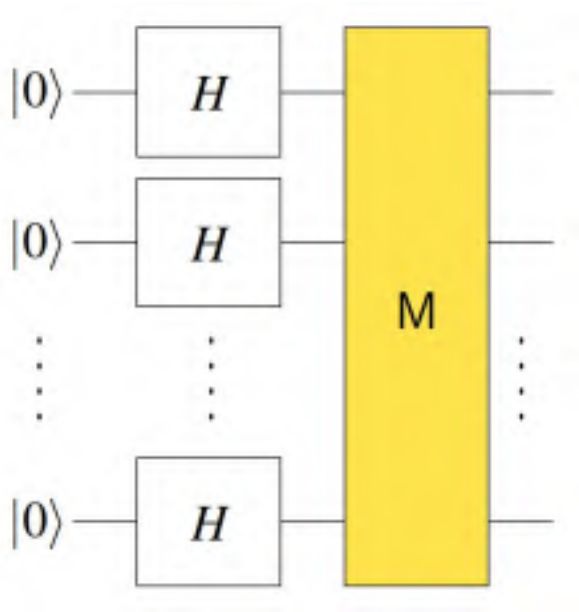
Tensorprodukt, Messen von Registern & Verschränkung

Mit den neuen Möglichkeiten können wir den Algorithmus zur Erzeugung eines Zufallsbits erweitern:

Algorithmus: n-Bit Zufallsgenerator

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

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



Tensorprodukt, Messen von Registern & Verschränkung

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

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

Für das Produkt zweier Faktoren beobachten wir

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

Tensorprodukt, Messen von Registern & Verschränkung

Insgesamt liefert die Produktbildung

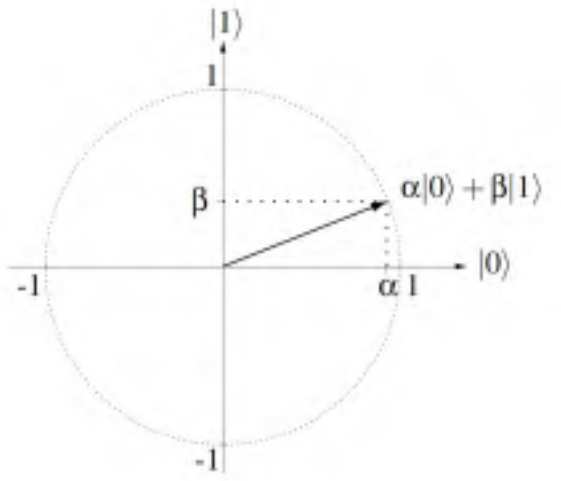
$$\frac{1}{\sqrt{2^n}}\left(|0...00\rangle + |0...01\rangle + ... + |1...1\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle.$$

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

Tensorprodukt, Messen von Registern & Verschränkung

Auch unser bisheriges Verständnis des *Messung* eines Quantenbits war ein Spezialfall, der immer auf ein klassisches Bit geführt hat.

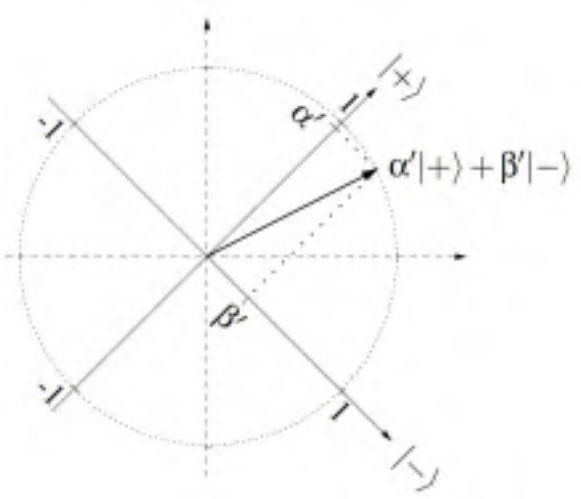
Dabei wurde ein Quantenbit $\alpha|0\rangle + \beta|1\rangle$ $(\alpha, \beta \in \mathbb{C})$ bzgl. der Standardbasis $|0\rangle, |1\rangle$ dargestellt.



Tensorprodukt, Messen von Registern & Verschränkung

Es können auch andere Basen gewählt werden, etwa

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



Dabei ändert sich der Zustandsvektor nicht, sondern nur dessen Projektion auf die Koordinatenachse.

Übung: Zeigen Sie, dass $|+\rangle, |-\rangle$ eine Orthogonalbasis ist.

Tensorprodukt, Messen von Registern & Verschränkung

Der Übergang

$$\alpha|0\rangle + \beta|1\rangle \rightsquigarrow \alpha'|+\rangle + \beta'|-\rangle$$

bleibt nicht ohne Konsequenzen.

Messen wir bezüglich der Basis $|+\rangle, |-\rangle$, so beobachten wir

- $|+\rangle$ mit Wahrscheinlichkeit $|\alpha'|^2$, und
- $|-\rangle$ mit Wahrscheinlichkeit $|\beta'|^2$.

Wie zuvor können wir nicht die Superposition als Ganzes ermitteln und die Superposition wird bei der Messung zerstört.

Tensorprodukt, Messen von Registern & Verschränkung

Um α', β' konkret zu bestimmen, müssen wir die Gleichung

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

lösen.

Übung: Zeigen Sie durch lösen der Gleichung

$$\alpha' = \frac{1}{\sqrt{2}}(\alpha + \beta), \quad \beta' = \frac{1}{\sqrt{2}}(\alpha - \beta).$$

Tensorprodukt, Messen von Registern & Verschränkung

Die bisherigen Überlegungen können wir allgemeiner formulieren:

Sei R ein Register aus n Quantenbits, das sich im Zustand $|\phi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$ befinde. Die orthogonalen Vektoren $|0'\rangle, |1'\rangle, ..., |(2^n-1)'\rangle$ 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 $|i'\rangle$.

Tensorprodukt, Messen von Registern & Verschränkung

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 $|\alpha_{00}|^2 + |\alpha_{01}|^2$.

Tensorprodukt, Messen von Registern & Verschränkung

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.

Tensorprodukt, Messen von Registern & Verschränkung

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

Tensorprodukt, Messen von Registern & Verschränkung

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

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

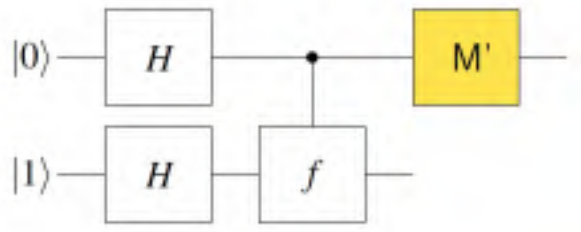
Bestimmen Sie für das zweite Qubit die Wahrscheinlichkeiten mit denen $|0\rangle$ bzw. $|1\rangle$ angenommen wird, sowie die Zustände.

Tensorprodukt, Messen von Registern & Verschränkung

Bezeichnet M' die Messung bzgl. $\{|+\rangle, |-\rangle\}$, dann ergibt sich eine alternative Version des Algorithmus für das Problem von Deutsch.

Algorithmus: Problem von Deutsch (Alternative Version)

1. $R = |xy\rangle \leftarrow |01\rangle$
2. $|x\rangle|y\rangle \leftarrow H_2 R$
3. $|x\rangle|y\rangle \leftarrow \mathrm{U_f}\, R$
4. Messe das erste Bit $|x\rangle$ bzgl. der Basis $\{|+\rangle, |-\rangle\}$:
   - $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$: f ist konstant
   - $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$: f ist balanciert



Tensorprodukt, Messen von Registern & Verschränkung

Mittels Tensorprodukt kann auch eine wesentliches Merkmal der Quantenmechanik beschrieben werden: die Verschränkung.

Beispiel: Auf das Register $|b_1 b_2\rangle$ im Zustand $|00\rangle$ wird zuerst die Operation $H \otimes I_2$ und dann CNOT : $|xy\rangle \mapsto |x, y \oplus x\rangle$ angewandt:

$$|00\rangle \xrightarrow{H\otimes I_2} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$
$$\xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Tensorprodukt, Messen von Registern & Verschränkung

Wird das erste Bit gemessen, ist das Ergebnis $|0\rangle$ oder $|1\rangle$, jeweils mit Wahrscheinlichkeit $1/2$.

- **1.Fall** Wird $|0\rangle$ gemessen, ist der Folgezustand $|00\rangle$.
- **2.Fall** Wird $|1\rangle$ gemessen, ist der Folgezustand $|11\rangle$.

Wird das zweite Bit gemessen, ergeben sich wieder $|0\rangle$ oder $|1\rangle$, jeweils mit Wahrscheinlichkeit $1/2$.

- **1.Fall** Wird $|0\rangle$ gemessen, ist der Folgezustand $|00\rangle$.
- **2.Fall** Wird $|1\rangle$ gemessen, ist der Folgezustand $|11\rangle$.

Tensorprodukt, Messen von Registern & Verschränkung

Wir beobachten:

Bevor $|b_1\rangle$ gemessen wurde, war der Ausgang der Messung an $|b_2\rangle$ noch offen, d.h. beide Ergebnisse waren gleich wahrscheinlich.

Ist $|b_1\rangle$ (bzw. $|b_2\rangle$) schon gemessen worden, steht das Ergebnis an $|b_2\rangle$ (bzw. $|b_1\rangle$) fest. Man sagt, die Zustände sind verschränkt.

Tensorprodukt, Messen von Registern & Verschränkung

Definition (Verschränkung)

Sei $|\phi\rangle$ der Zustand eines Quantenregisters aus n Bits. Der Zustand $|\phi\rangle$ heißt *unverschränkt*, wenn er das Produkt von Zuständen der einzelnen Bits ist:

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

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

Tensorprodukt, Messen von Registern & Verschränkung

Beispiel:

$$\begin{aligned} H_2|11\rangle &= \frac{1}{2}\left(|00\rangle - |01\rangle - |10\rangle + |11\rangle\right) \\ &= \frac{1}{2}\left(|0\rangle \otimes \left(|0\rangle - |1\rangle\right) - |1\rangle \otimes \left(|0\rangle - |1\rangle\right)\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \\ &= H|1\rangle \otimes H|1\rangle \end{aligned}$$

Tensorprodukt, Messen von Registern & Verschränkung

Übung: Die Zustände

$$\Phi^+ = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad \Phi^- = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$
$$\Psi^+ = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \quad \Psi^- = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

heißen *Bell-Zustände*. Zeigen Sie, dass Φ^+ nicht in das Produkt zweier Zustände jeweils eines Bits zerlegt werden kann.

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

Tensorprodukt, Messen von Registern & Verschränkung

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

Man kann (formal) ein Maß an Verschränkung definieren. Dabei lässt sich beobachten, dass die Bell-Zustände *maximal verschränkt* sind. Andererseits ist ein Zustand der Form

$$\frac{1}{\sqrt{k}}|00\rangle + \sqrt{\frac{k-1}{k}}|11\rangle$$

für wachsendes $k \geq 2$ „immer weniger verschränkt“.

Tensorprodukt, Messen von Registern & Verschränkung

Zum Abschluss des Abschnitts soll noch einmal die Hadamard-Transformation aufgegriffen werden:

Aus der Analyse des n-Bit Zufallsgenerators ist

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

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

Tensorprodukt, Messen von Registern & Verschränkung

Wir untersuchen die Wirkung von H_n auf ein Register $R = |x_{n-1}...x_0\rangle$ mit $x_i \in \{0,1\}$ für $i = 0,...,n-1$ und beginnen mit dem Fall $n = 2$:

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

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

Tensorprodukt, Messen von Registern & Verschränkung

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

wenn $\cdot$ das Skalarprodukt zweier Vektoren bezeichnet.

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

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

Tensorprodukt, Messen von Registern & Verschränkung

Es entsteht eine Superposition über alle klassischen Zustände des Registers. Die Information über $|\mathbf{x}\rangle$ wird in die Amplitude verlagert.

Beispiel: Als *gleichgewichtete* Superposition bezeichnet man

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

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

$$H_n|0...01\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{y_0} |y\rangle,$$

Tensorprodukt, Messen von Registern & Verschränkung

wobei

$$\begin{aligned}\sum_{y=0}^{2^n-1}(-1)^{y_0}|y\rangle &= (-1)^0|0\ldots00\rangle + (-1)^1|0\ldots01\rangle + \\ &\quad + (-1)^0|0\ldots010\rangle + (-1)^1|0\ldots011\rangle + \ldots \\ &\quad + (-1)^0|1\ldots10\rangle + (-1)^1|1\ldots11\rangle \\ &= |0\ldots00\rangle - |0\ldots01\rangle + |0\ldots010\rangle - |0\ldots011\rangle + \ldots \\ &\quad + |1\ldots10\rangle - |1\ldots11\rangle.\end{aligned}$$

Wir werden dies im nächsten Abschnitt verwenden.

Inhaltsverzeichnis

Das RSA-Verfahren & periodische Funktionen

Ausgangssituation: Alice möchte Bob etwas Wichtiges mitteilen. Eve interessiert sich ebenfalls für den Inhalt der Nachricht. Es soll verhindert werden das Eve den Inhalt der Nachricht erfährt.

Public Key Kryptographie: Bob erwartet eine vertrauliche Nachricht.

1. Bob erzeugt zwei Schlüssel: einen geheimen, den er für sich behält, und einen öffentlichen, den er Alice zukommen lässt.
2. Alice verschlüsselt die Nachricht an Bob mit dem öffentlichen Schlüssel (von Bob).
3. Bob verwendet seinen geheimen Schlüssel um die Nachricht von Alice zu entschlüsseln.

Das RSA-Verfahren & periodische Funktionen

Schlüsselerzeugung bei RSA

1. Wähle zwei Primzahlen $p \neq q$.
2. Berechne $n = pq$ und $\phi(n) = (p-1)(q-1)$.
3. Wähle eine kleine Zahl e mit $ggT(e, \phi(n)) = 1$.
4. Löse $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Das Paar (e, n) ist der *öffentlicher Schlüssel* und das Paar (d, n) der *geheime Schlüssel*.

Für jeden der oben genannten Schritte gibt es effiziente Algorithmen.

Das RSA-Verfahren & periodische Funktionen

Ver- und Entschlüsselung bei RSA

- Verschlüsselung einer Nachricht m: Löse $c \equiv m^e \pmod{n}$
- Entschlüsselung einer Nachricht c: Löse $m \equiv c^d \pmod{n}$

Beobachtung: Das Paar (e, n) ist als öffentlicher Schlüssel auch für Eve zugänglich; ebenso die übertragene Nachricht. Um den Klartext m zu bestimmen, benötigt Eve den geheimen Schlüssel d und muss dafür

$$ed \equiv 1 \pmod{(p-1)(q-1)}$$

lösen. Dafür benötigt Eve das Produkt $(p-1)(q-1) = \phi(n)$. Sie kann versuchen $n = pq$ zu faktorisieren.

Das RSA-Verfahren & periodische Funktionen

Aber: Ist n groß genug gewählt, dann ist die Faktorisierung von n (bisher) nicht effizient möglich.
Natürlich wählt Bob n von entsprechender Größe.

Verfügt Eve über einen Quantencomputer (mit ausreichend vielen Qubits), dann kann Sie n effizient faktorisieren

$\rightsquigarrow$ *Algorithmus von Shor* (Perioden von Funktionen)

Das RSA-Verfahren & periodische Funktionen

Definition (Periode einer Funktion)

Sei $f : \mathbb{R} \to \mathbb{R}$ eine Funktion und $p \in \mathbb{R}^+$ die kleinste Zahl, sodass $f(x+p) = f(x)$ für alle $x \in \mathbb{R}$ gilt. Dann heißt p die *Periode* von f.

Beispiel:

- $\sin(x + 2\pi) = \sin(x)$
- $f(x) = 2^x \pmod 5$ hat die Periode $p = 4$, denn $f(0) = 1, f(1) = 2, f(2) = 4, f(3) = 3, f(4) = 1, \ldots$

Das RSA-Verfahren & periodische Funktionen

Ziel: n $(= pq)$ aus dem RSA-Verfahren faktorisieren, d.h. einen echten Teiler $1 < a < n$ bestimmen.

O.B.d.A. kann n als ungerade angenommen werden (sonst sind 2 und $n/2$ echte Teiler).

Randomisierter Ansatz (so nicht)

- Idee: nutze den Euklidischen Algorithmus
- wähle zufällig $a \in \{2, ..., n-1\}$ und bestimme einen Teiler durch Berechnung des $ggT(a, n)$
- dabei wird oft 1 als gemeinsamer Teiler auftreten, denn $n = pq$ ist das Produkt *großer* Primzahlen p, q und $\phi(n) = \#\{1 \leq a \leq n | ggT(a, n) = 1\} = (p-1)(q-1)$; d.h. *fast alle* Zahlen $\{2, ..., n-1\}$ sind teilerfremd zu n

Das RSA-Verfahren & periodische Funktionen

Angenommen wir verfügen über folgende

Basisfähigkeit: Für $1 < a < n$ können wir die Periode p von $f(x) = a^x \pmod{n}$ bestimmen.

Da p die Periode ist, folgt

$$\begin{aligned} f(x+p) = f(x) &\Rightarrow f(p) = f(0) \\ &\Rightarrow a^p \equiv 1 \pmod{n} \\ &\Rightarrow a^p = 1 + kn \quad (k \in \mathbb{Z}). \end{aligned}$$

Also ist n ein Teiler von $a^p - 1$.

Das RSA-Verfahren & periodische Funktionen

Nehmen wir weiter an das die Periode p gerade ist, folgt

$$a^p - 1 = (a^{p/2} - 1)(a^{p/2} + 1) = kn.$$

Daraus folgt $ggT\left(a^{p/2} - 1, n\right) > 1$ oder $ggT\left(a^{p/2} + 1, n\right) > 1$.

Denn: Angenommen $ggT\left(a^{p/2} - 1, n\right) = ggT\left(a^{p/2} + 1, n\right) = 1$.
Dann sind wegen der Eindeutigkeit der Primfaktorzerlegung die Faktoren $a^{p/2} \pm 1$ *beide Teiler von* k, *also*

$$(a^{p/2} - 1)(a^{p/2} + 1) = kn = k'(a^{p/2} - 1)(a^{p/2} + 1)n$$

für $k' \in \mathbb{Z}$. *Insbesondere folgt* $n = 1$, *im Widerspruch zu* $n > 1$.

Das RSA-Verfahren & periodische Funktionen

Also gilt $ggT\left(a^{p/2}-1,n\right)>1$ oder $ggT\left(a^{p/2}+1,n\right)>1$.

Das beinhaltet auch die Möglichkeit das $a^{p/2}+1$ ein Vielfaches von n ist. Dann ist $ggT\left(a^{p/2}+1,n\right)=n$ und wir erhalten keine Information.

Übung: Zeigen Sie, dass $a^{p/2}-1$ kein Vielfaches von n ist.

Damit ist $ggT\left(a^{p/2}-1,n\right)$ oder $ggT\left(a^{p/2}+1,n\right)$ ein echter Teiler von n, außer

- $ggT\left(a^{p/2}-1,n\right)=1$, und
- $a^{p/2}+1$ ist ein Vielfaches von n.

Das RSA-Verfahren & periodische Funktionen

Proposition

Sei n ungerade und keine Primzahlpotenz. Dann gilt für mindestens die Hälfte der Fälle der Zahlen $0 \leq a \leq n-1$ mit $ggT(a,n) = 1$

1. die Periode p der Funktion $f(x) = a^x \pmod{n}$ ist gerade,
2. n teilt nicht $a^{p/2}+1$, d.h. $a^{p/2} \not\equiv -1 \pmod{n}$.

Unsere Überlegungen können wir als Algorithmus festhalten. Dabei haben wir (bisher) noch keinen Quantencomputer eingesetzt.

Das RSA-Verfahren & periodische Funktionen

Faktorisierungsalgorithmus - klassischer Teil

Eingabe: Eine ungerade ganze Zahl n, die keine Primzahlpotenz ist.
Ausgabe: Ein echter Teiler von n.

1. Wähle zufällig eine Zahl $a \in \{2, ..., n-1\}$.
2. $z \leftarrow ggT(a, n)$.
   Falls $z > 1$: Ausgabe von z. Abbruch
3. Ermittle die Periode p von $a^x \pmod{n}$.
4. Falls p ungerade ist: Beginne erneut mit *Schritt 1*.
5. Ermittle $ggT\left(a^{p/2} - 1, n\right)$ und $ggT\left(a^{p/2} + 1, n\right)$. Hat sich kein echter Teiler ergeben, beginne erneut mit *Schritt 1*.
   Sonst: Ausgabe z.

Das RSA-Verfahren & periodische Funktionen

Schritt 3 des Algorithmus werden wir (später) mit einem Quantenalgorithmus lösen $\rightsquigarrow$ *Shor-Algorithmus*

Fehlerwahrscheinlichkeit

Ist n ungerade und nicht die Potenz einer Primzahl, d.h. $n \neq p^k$ für $p \in \mathbb{P}$ und $k \geq 1$. Dann führt der klassische Teil des Algorithmus mit einer Wahrscheinlichkeit größer $1/2$ im ersten Anlauf zum Erfolg.

Die Wahrscheinlichkeit, nach k Versuchen noch keine Lösung gefunden zu haben, ist demnach kleiner $1/2^k$.

Das RSA-Verfahren & periodische Funktionen

Mit $ggT(x, n) = ggT(x \pmod{n}, n)$ für $x, n \in \mathbb{Z}$ folgt

Laufzeitbetrachtung:

- Schritt 1: zufällige Auswahl: konstante Laufzeit $\rightsquigarrow O(1)$
- Schritt 2: Euklidischer Algorithmus $\rightsquigarrow O\left((\log n)^3\right)$
- Schritt 3: Periodenbestimmung: unbekannte Laufzeit $T(n)$
- Schritt 4: prüfe gerade, ungerade: konstante Laufzeit $\rightsquigarrow O(1)$
- Schritt 5: Euklidischer Algorithmus $\rightsquigarrow O\left((\log n)^3\right)$

Das RSA-Verfahren & periodische Funktionen

Wir wissen bereits, dass die Wahrscheinlichkeit nach k Versuchen keine Lösung gefunden zu haben, kleiner als $1/2^k$ ist. Die erwartete Laufzeit ist also

$$\sum_{k \geq 1} \frac{k}{2^k} \cdot O\left((\log n)^3\right) = O\left((\log n)^3\right).$$

Übung: Zeigen Sie die Konvergenz der Reihe $\sum_{k \geq 1} \frac{k}{2^k}$.

Das RSA-Verfahren & periodische Funktionen

Gesamtlaufzeit

Sei n ungerade und nicht die Potenz einer Primzahl, d.h. $n \neq p^k$ für $p \in \mathbb{P}$ und $k \geq 1$. Ein echter Teiler von n kann mit einer Laufzeit von

$$O\left((\log n)^3\right) + T(n)$$

bestimmt werden, wenn $T(n)$ die Laufzeit zur Bestimmung einer Periode der Funktion $a^x \pmod{n}$ für gegebenes $a \in \{2, \ldots, n-1\}$ bezeichnet.

Frage: Wie bestimmt man die Periode?

Schnelle Fouriertransformation

Aufgabe: Multiplikation von Polynomen

Definition: Koeffizientendarstellung

Sei $A(x) = \sum_{i=0}^{n-1} a_i x^i \in \mathbb{C}[x]$ ein Polynom vom Grad $n-1$. Dann heißt $(a_0, ..., a_{n-1}) \in \mathbb{C}^n$ die *Koeffizientendarstellung* von A.

Proposition: Multiplikation („Schulmethode“)

Seien $A(x) = \sum_{i=0}^{n-1} a_i x^i$, $B(x) = \sum_{i=0}^{n-1} b_i x^i \in \mathbb{C}[x]$ Polynome.
Dann gilt für das Produkt der Polynome
$A(x) \cdot B(x) = C(x) = \sum_{i=0}^{2n-2} c_i x^i$, wobei $c_i = \sum_{j=0}^{i} a_j b_{i-j}$.
Die Berechnung benötigt eine Laufzeit von $O(n^2)$.

Schnelle Fouriertransformation

Wir wollen uns überlegen wie das schneller geht:

Proposition (Stützstellendarstellung, Interpolationspolynom)

Seien $(x_0, y_0), \ldots, (x_{n-1}, y_{n-1}) \in \mathbb{C}^2$ n paarweise verschiedene Punkte. Dann gibt es genau ein Polynom A vom Grad $n-1$ mit

$$A(x_i) = y_i, \quad 1 \leq i \leq n-1.$$

Die Punkte (x_i, y_i) heißen *Stützstellen* von A und $(x_0, y_0), \ldots, (x_{n-1}, y_{n-1})$ die *Stützstellendarstellung* von A. Das Polynom A nennt man *Interpolationspolynom*.

Schnelle Fouriertransformation

Das lässt sich auch mit der *Vandermonde Matrix* $V(x)$ beschreiben:

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{pmatrix}}_{=V(x)} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}.$$

Die Vandermonde Matrix ist invertierbar, d.h. die Koeffizientendarstellung $\mathbf{a} = (a_0, ..., a_{n-1})^T$ lässt sich eindeutig aus $\mathbf{y} = (y_0, ..., y_{n-1})^T$ berechnen: $\mathbf{a} = V(x)^{-1}\mathbf{y}$.

Schnelle Fouriertransformation

Für uns sind die folgende Beobachtungen interessant:

- Ist $(x_0, A(x_0))$ Stützstelle von A und $(x_0, B(x_0))$ Stützstelle von B, dann ist $(x_0, A(x_0)B(x_0))$ Stützstelle des Produktes C, wobei $C(x) = A(x)B(x)$.
- Sind A und B jeweils vom Grad $n-1$, dann hat C den Grad $2n-2$; d.h. wir benötigen $2n-1$ Stützstellen.
- Die Berechnung der Stützstellen dieses Produktes benötigt eine Laufzeit von $O(n)$.

Schnelle Fouriertransformation

Zusammen:

- Schulmultiplikation $\rightsquigarrow$ Laufzeit $O(n^2)$
- punktweise Multiplikation mittels Stützstellen $\rightsquigarrow$ Laufzeit $O(n)$
- Laufzeiten L_1 (punktweise Auswertung) und L_2 (Interpolation) sind uns (noch) unbekannt
  Ziel: L_1, L_2 in Laufzeit $O(n \log n)$

$$\begin{array}{ccc} \text{Koeffizientendarstellung} & \xrightarrow{O(n^2)} & \text{Koeffizientendarstellung} \\ \Big\downarrow L_1 & & \Big\uparrow L_2 \\ \text{Stützstellendarstellung} & \xrightarrow{O(n)} & \text{Stützstellendarstellung} \end{array}$$

Schnelle Fouriertransformation

Definition (Einheitswurzel)

Die (komplexen) Lösungen der Gleichung $x^n = 1$ heißen n-te *Einheitswurzeln.*

Es gibt genau n Einheitswurzeln. Diese sind die Potenzen von

$$\omega_n = \exp\left(\frac{2\pi i}{n}\right),$$

also $1 = \omega_n^0, \omega_n, \omega_n^2, ..., \omega_n^{n-1}$, wobei $\omega_n^n = \exp(2\pi i) = 1$.

Schnelle Fouriertransformation

Polynome lassen sich an Einheitswurzeln leicht auswerten. Auch Zwischenergebnisse solcher Auswertungen können weiter verwendet werden (etwa $\omega_n \rightsquigarrow \omega_n^2$).

Definition (diskrete Fouriertransformation, DFT)

Sei A ein Polynom vom Grad $n-1$, gegeben in Koeffizientendarstellung $\mathbf{a} = (a_0, ..., a_{n-1})^T$. Dann heißt $\mathbf{y} = (y_0, ..., y_{n-1})^T$ mit

$$y_k = A(\omega_n^k), \quad 0 \leq k \leq n-1$$

diskrete Fouriertransformation von $\mathbf{a}$. Bezeichnung: $\mathbf{y} = DFT_n(\mathbf{a})$.

Schnelle Fouriertransformation

Die DFT eines Polynoms A vom Grad $n-1$ ist also dessen Stützstellendarstellung bzgl. der n-ten Einheitswurzel.

Die zu ω_n^k gehörende Komponente ist gegeben durch $y_k = \sum_{j=0}^{n-1} a_j \omega_n^{kj}$, bzw.

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Schnelle Fouriertransformation

Die angegeben Matrix ist invertierbar, d.h. auch DFT_n ist invertierbar und in der entsprechenden Matrix zu DFT_n^{-1} ist $\frac{1}{n}\omega_n^{-(j-1)(k-1)}$ der Eintrag an Stelle (j,k).

Wir betrachten die Einheitswurzeln genauer:

Beobachtung: Ist $n > 1$ gerade, dann sind die Quadrate n-ter Einheitswurzeln selbst $n/2$-te Einheitswurzeln. Also

$$\left\{1, (\omega_n)^2, (\omega_n^2)^2, ..., (\omega_n^{n-1})^2\right\} = \left\{1, \omega_{\frac{n}{2}}, \omega_{\frac{n}{2}}^2, ..., \omega_{\frac{n}{2}}^{\frac{n}{2}-1}\right\},$$

denn

$$(\omega_n^k)^2 = \exp\left(\frac{2\pi i}{n} \cdot 2k\right) = \exp\left(\frac{2\pi i}{n/2} \cdot k\right) = \omega_{\frac{n}{2}}^k \quad \forall k \in \mathbb{N}.$$

Schnelle Fouriertransformation

Übung Zeigen Sie $\{1, \omega_4^2, (\omega_4^2)^2, (\omega_4^3)^2\} = \{1, \omega_2\}$.

Es treten also Wiederholungen auf. Genauer: Durch quadrieren der n-ten Einheitswurzeln ergibt sich jede $n/2$-te Einheitswurzel genau zweimal.

Insbesondere halbiert sich die Anzahl der Einheitswurzeln, die man berechnen muss!

Schnelle Fouriertransformation

Es folgt

$$\begin{aligned} A(x) &= a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} \\ &= \left(a_0 + a_2 x^2 + a_4 x^4 + \ldots + a_{n-2} x^{n-2}\right) + \\ &\quad + x\left(a_1 + a_3 x^2 + a_5 x^4 + \ldots + a_{n-1} x^{n-2}\right) \\ &= A_0(x^2) + x A_1(x^2) \end{aligned}$$

für

$$\begin{aligned} A_0(x) &:= a_0 + a_2 x + a_4 x^2 + \ldots + a_{n-2} x^{\frac{n}{2}-1} \\ A_1(x) &:= a_1 + a_3 x + a_5 x^2 + \ldots + a_{n-1} x^{\frac{n}{2}-1}. \end{aligned}$$

Schnelle Fouriertransformation

Das Polynom A kann also an $\omega_n^0, \omega_n, \omega_n^2, ..., \omega_n^{n-1}$ ausgewertet werden, indem man die Polynome A_0 und A_1 vom halben Grad an den Einheitswurzeln $(\omega_n^0)^2, \omega_n^2, (\omega_n^2)^2, ..., (\omega_n^{n-1})^2$ auswertet.

Algorithmus FFT_n (Fast Fourier Transformation)

Eingabe: $(a_0, ..., a_{n-1})$, wobei $n = 2^k, k \in \mathbb{N}$
Ausgabe: $y = DFT_n(a_0, ..., a_{n-1})$

1. Falls $n - 1 = 0$. Ausgabe a_0.
2. $a^0 \leftarrow (a_0, a_2, ..., a_{n-2}), \quad a^1 \leftarrow (a_1, a_3, ..., a_{n-1})$
3. $y^0 \leftarrow FFT_{\frac{n}{2}}(a^0), \quad y^1 \leftarrow FFT_{\frac{n}{2}}(a^1)$
4. Setze y^0 und y^1 entsprechend $A(x) = A_0(x^2) + xA^1(x^2)$ zu $y = DFT_n(a)$ zusammen.

Schnelle Fouriertransformation

Bezeichnet $T(n)$ die Laufzeit von FFT_n, so ergibt sich die Rekursion

$$T(1) = O(1)$$
$$T(n) = 2T(n/2) + O(n)$$

Lösen der Rekursionsgleichung liefert $T(n) = O(n \log n)$.

Schnelle Fouriertransformation

Zusammen:

- Schulmultiplikation $\rightsquigarrow$ Laufzeit $O(n^2)$
- punktweise Multiplikation mittels Stützstellen $\rightsquigarrow$ Laufzeit $O(n)$
- Auswertung mittels FFT_n $\rightsquigarrow$ Laufzeit $O(n \log n)$
- Interpolation mittels FFT_n^{-1} $\rightsquigarrow$ Laufzeit $O(n \log n)$

$$\begin{array}{ccc} \text{Koeffizientendarstellung} & \xrightarrow{O(n^2)} & \text{Koeffizientendarstellung} \\ \Big\downarrow {\scriptstyle O(n \log n)} & & \Big\uparrow {\scriptstyle O(n \log n)} \\ \text{Stützstellendarstellung} & \xrightarrow{O(n)} & \text{Stützstellendarstellung} \end{array}$$

Quanten-Fouriertransformation

Frage: Lassen sich die Ergebnisse zur DFT_n auf einem Quantencomputer realisieren?

Definition (Quanten-Fouriertransformation - Teil I)

Die *Quanten-Fouriertransformation* der Ordnung N ist durch die unitäre Matriz

$$QFT_N := \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{pmatrix}$$

festgelegt, wobei ω_N die N-te Einheitswurzel $\exp\left(\frac{2\pi i}{N}\right)$ bezeichnet.

Quanten-Fouriertransformation

Definition (Quanten-Fouriertransformation - Teil II)

Ist $|0\rangle, |1\rangle, ..., |N-1\rangle$ eine Orthonormalbasis, dann operiert QFT_N auf dem N-dimensionalen Basisvektor $|j\rangle$ vermöge

$$QFT_N|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{jk} |k\rangle.$$

Für einen beliebigen Zustandsvektor $|v\rangle = \sum_{j=0}^{N-1} \alpha_j |j\rangle$ gilt also

$$QFT_N|v\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(\sum_{j=0}^{N-1} \alpha_j \omega_N^{jk} \right) |k\rangle.$$

Quanten-Fouriertransformation

Der Zusammenhang zwischen *DFT* und *QFT* wird erkennbar, wenn man sich die Amplituden als Koeffizienten eines Polynoms vorstellt.

Zu klären bleibt, ob sich die *QFT* auch effizient ausführen lässt. Unser Ziel ist

Satz

Für $N = 2^n$ lässt sich QFT_N mit $O(n^2)$ vielen Quantengattern realisieren.

Wir werden uns die Beweisidee anhand eines Beispiels verdeutlichen.

Quanten-Fouriertransformation

Für $N = 4$ ist

$$\begin{aligned} QFT_4|x\rangle &= \frac{1}{2}\sum_{y=0}^{3}\omega_4^{xy}|y\rangle \\ &= \frac{1}{2}\left(\omega_4^0|00\rangle + \omega_4^x|01\rangle + \omega_4^{2x}|10\rangle + \omega_4^{3x}|11\rangle\right) \\ &= \frac{1}{2}\left(|0\rangle + \omega_4^{2x}|1\rangle\right)\cdot\left(|0\rangle + \omega_4^x|1\rangle\right) \\ &= \frac{1}{2}\left(|0\rangle + \omega_2^x|1\rangle\right)\cdot\left(|0\rangle + \omega_4^x|1\rangle\right), \end{aligned}$$

wobei wir im ersten Faktor $\omega_4^{2x} = \exp\left(\frac{2\pi i\cdot 2x}{4}\right) = \exp\left(\frac{2\pi i\cdot x}{2}\right) = \omega_2^x$ verwenden.

Quanten-Fouriertransformation

Allgemeiner gilt:

Lemma

Besteht der Zustand $|x\rangle$ aus n Bits, wobei $N = 2^n$ gelte. Dann ist

$$QFT_N|x\rangle = \frac{1}{\sqrt{N}}\left(|0\rangle + \omega_2^x|1\rangle\right)\cdot\left(|0\rangle + \omega_4^x|1\rangle\right)\ldots\left(|0\rangle + \omega_N^x|1\rangle\right).$$

Übung: Zeigen Sie die Gültigkeit der Gleichung für $N = 8$.

Quanten-Fouriertransformation

Zur Umsetzung mittels einem Quantenregister beobachten wir

$$H|x\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + \omega_2^x|1\rangle\right).$$

2-te Einheitswurzeln alleine sind jedoch nicht ausreichen.

Definition (Phasen-Rotation)

Sei $m \geq 4$. Die durch

$$R_m := \begin{pmatrix} 1 & 0 \\ 0 & \omega_m \end{pmatrix}$$

beschriebene Operation heißt *Phasen-Rotation*.

Quanten-Fouriertransformation

Übung:

1. Berechne $R_4 \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$.
2. Zeigen Sie, dass R_m unitär ist.

Um ω_N^x zu berechnen, beobachten wir:

- Für den ganzzahligen Wert des Bitvektors $x = (x_2 x_1 x_0)_2$ gilt $x \mod 2 = x_0$, $x \mod 4 = 2x_1 + x_0$ und $x \mod 8 = 4x_2 + 2x_1 + x_0$.
- $\omega_N^k = \omega_N^{k \bmod N}$ (da die komplexe Exponentialfunktion periodisch ist).

Quanten-Fouriertransformation

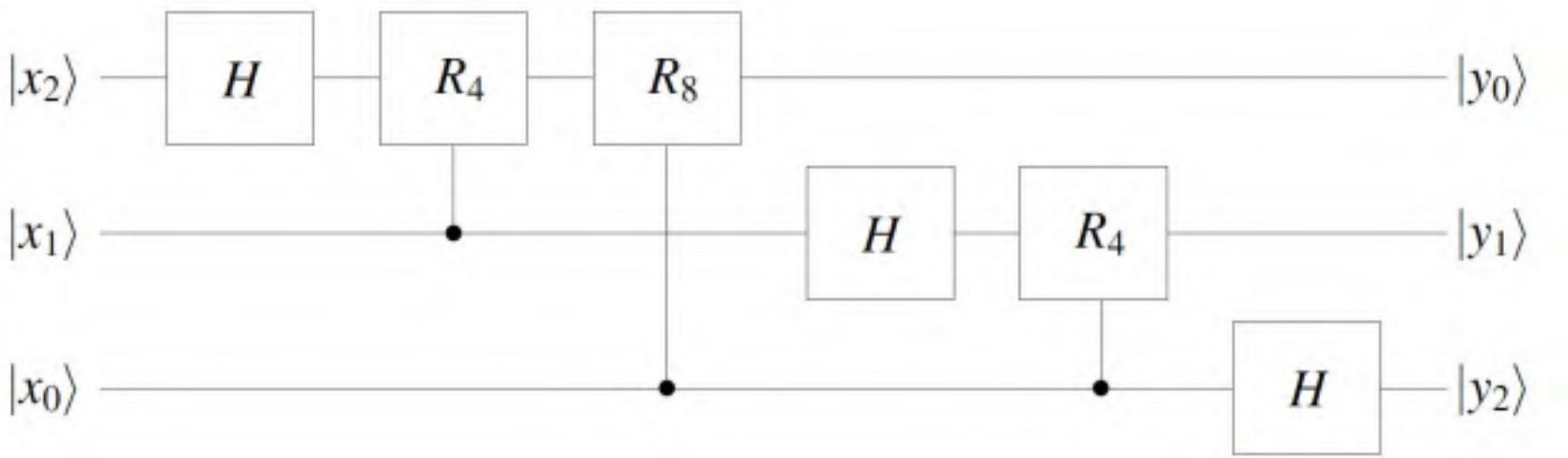
Es folgt

- $\omega_2^x = \omega_2^{x \pmod 2} = \omega_2^{x_0}$ (lässt sich Hadamard-Transformation realisieren)
- $\omega_4^x = \omega_4^{2x_1+x_0} = \omega_4^{2x_1}\omega_4^{x_0} = \omega_2^{x_1}\omega_4^{x_0}$
- $\omega_8^{4x_2+2x_1+x_0} = \omega_8^{4x_2}\omega_8^{2x_1}\omega_8^{x_0} = \omega_2^{x_2}\omega_4^{x_1}\omega_8^{x_0}$

Man kann also $\omega_m^{x_i}$ mit einem durch x_i gesteuerten Phasen-Rotationsgatter bestimmen.

Quanten-Fouriertransformation

Graphische Darstellung von QFT_8:



Die Ideen aus dem Beispiel lassen sich verallgemeinern, sodass tatsächlich $O(n^2)$ viele Quantengatter ausreichen.

Simons- & Shors Algorithmus

Zur Vorbereitung auf den Algorithmus von Shor betrachten wir zunächst den strukturell ähnlich *Algorithmus von Simon*:

Ausgangssituation: Gegeben ist eine Funktion $f : \{0,1\}^n \to \{0,1\}^n$ für die eine der Alternativen gilt:

1. f ist bijektiv
2. Je zwei Vektoren x, x' aus $\{0,1\}^n$ haben dasselbe Bild und es gibt ein $s \in \{0,1\}^n,\ s \neq 0$ mit $f(x) = f(x')$ genau dann, wenn $x \oplus s = x'$.

Aufgabe: Entscheide, welcher Fall vorliegt. Im zweiten Fall soll s angeben werden.

Simons- & Shors Algorithmus

Wegen $f(x \oplus s) = f(x)$ wird s als Periode von f bezeichnet.

Beispiel: Für die Funktion
$f : \{0,1\}^n \to \{0,1\}^n,\ f(x,y) = (0, x \oplus y)$ gilt

$$00 \mapsto 00, \quad 01 \mapsto 01, \quad 10 \mapsto 01, \quad 11 \mapsto 00.$$

Beobachtung:

- 00 und 11 werden auf 00 abgebildet,
- 01 und 10 werden auf 11 abgebildet.

Die Periode ist also $s = (1,1)$.

Simons- & Shors Algorithmus

Simons Algorithmus

$|a\rangle|b\rangle$ zwei n-Bit Quantenregister
Eingabe: Quantenorakel $U_f : |a\rangle|b\rangle \mapsto |a\rangle|b \oplus f(a)\rangle$

1. $R = |a\rangle|b\rangle \leftarrow |0...0\rangle|0...0\rangle$
2. Wende Hadamard-Transformation H_n auf $|a\rangle$ an:

$$R \leftarrow H_n|a\rangle|b\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0...0\rangle$$

3. Wende Orakel U_f an:

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

Simons- & Shors Algorithmus

Simons Algorithmus

4. Messe das Register $|b\rangle$
5. Wende die Hadamard-Transformation H_n auf $|a\rangle$ an: $|a\rangle \leftarrow H_n|a\rangle$
6. Messe das Register $|a\rangle$. Das Ergebnis ist ein Vektor z, der ausgegeben wird.

Wiederhole diesen Prozess, bis $n-1$ linear unabhängige Vektoren $z_1, \ldots, z_{n-1}$ erzeugt wurden.

Die Schritte 1.-6. bilden den Quantenteil von Simons Algorithmus. Die kommenden Schritte 7. & 8. bilden den klassischen Teil.

Simons- & Shors Algorithmus

Simons Algorithmus

7. Löse das lineare Gleichungssystem

$$z_1 t = 0, ..., z_{n-1} t = 0$$

in den Unbekannten $t = (t_1, ..., t_{n-1})$. Die Lösung, ungleich dem Nullvektor, sei s.

8. Ist $f(s) = f(0)$, dann Ausgabe: f ist periodisch mit Periode s. Sonst, Ausgabe: f ist bijektiv.

Simons- & Shors Algorithmus

Analyse des Algorithmus: Die Anwendung von U_f erzeugt eine Verschränkung der Register $|a\rangle$ und $|b\rangle$. In **Schritt 4** können zwei Möglichkeiten auftreten:

1. Möglichkeit: Ist f *bijektiv* und ergibt die Messung von $|b\rangle$ den Wert y, dann enthält $|a\rangle$ den Wert $f^{-1}(y)$.

2. Möglichkeit: Ist f *nicht bijektiv* und ergibt die Messung von $|b\rangle$ den Wert y, dann hat y zwei Urbilder x und x' mit $x \oplus s = x'$. Die Messung beider Register hätte entweder $|x\rangle|f(x)\rangle$ oder $|x'\rangle|f(x)\rangle = |x \oplus s\rangle|f(x)\rangle$ ergeben.

Simons- & Shors Algorithmus

Da nur das zweite Register gemessen wird, ist R nach **Schritt 4** im Zustand

$$|\phi_4\rangle = \frac{1}{\sqrt{2}} \left(|\hat{x}\rangle + |\hat{x} \oplus s\rangle \right) |f(\hat{x})\rangle.$$

Da $f(\hat{x})$ gleichverteilt gewählt wurde, könnte $|\hat{x}\rangle$ jeder Zustandsvektor sein. Um $\hat{x}$ aus $|\phi_4\rangle$ zu entfernen, wird in **Schritt 5** H_n angewandt und liefert

$$\begin{aligned} |\phi_5\rangle &= \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} \frac{1}{\sqrt{2}} \left((-1)^{\hat{x} \cdot z} + (-1)^{(\hat{x} \oplus s) \cdot z} \right) |z\rangle |f(\hat{x})\rangle \\ &= \sum_{z \in \{0,1\}^n} \frac{1}{\sqrt{2^{n+1}}} \left((-1)^{\hat{x} \cdot z} + (-1)^{\hat{x} \cdot z \oplus s \cdot z} \right) |z\rangle |f(\hat{x})\rangle. \end{aligned}$$

Simons- & Shors Algorithmus

Für die Amplitude α_z können zwei Fälle eintreten:

- $s \cdot z$ ist gerade. Dann ist

$$\alpha_z = \frac{1}{\sqrt{2^{n+1}}} \left((-1)^{\hat{x} \cdot z} + (-1)^{\hat{x} \cdot z} \right) = \pm \frac{1}{\sqrt{2^{n-1}}}.$$

- $s \cdot z$ ist ungerade. Dann ist

$$\alpha_z = \frac{1}{\sqrt{2^{n+1}}} \left((-1)^{\hat{x} \cdot z} + (-1)^{\hat{x} \cdot z \oplus 1} \right) = 0.$$

Der Einfluss von $\hat{x}$ konnte also durch Hadamard-Transformation beseitigt werden.

Simons- & Shors Algorithmus

Die letzte Messung in **Schritt 6** liefert dann einen Vektor z mit

$$z \cdot s \equiv 0 \pmod 2.$$

Tatsächlich gibt es genau 2^{n-1} solcher Vektoren z_j.

Kennen wir schon $n-1$ dieser Vektoren, kann in **Schritt 7** das lineare Gleichungssystem gelöst werden. Insbesondere gibt es eine nichttriviale Lösung $s \neq 0$.

Ist f nicht periodisch, dann sind die z_j gleichverteilte Zufallsvektoren und damit ist auch die Lösung s ein Zufallsvektor. Dann gilt $f(0) \neq f(s)$ und f ist *bijektiv*.

Simons- & Shors Algorithmus

Zur Laufzeit des Algorithmus lässt sich noch bemerken

- Mit Wahrscheinlichkeit $1/4$ sind schon die ersten $n-1$ Vektoren, die vor Schritt 7 zu ermitteln sind, linear unabhängig.
- Lösen des linearen Gleichungssystems mittels Gaußschem Eliminationsverfahren benötigt $O(m^3)$ Schritte.

Der Algorithmus von Simon ist also ein Polynomialzeitalgorithmus.

Simons- & Shors Algorithmus

Beispiel: Betrachte

$$f : \{0,1\}^3 \to \{0,1\}^3,\ f(x_1, x_1, x_0) = (x_2, x_1, 0)$$

Wir wollen zeigen das die Periode $s = (0,0,1)^T$ ist.

Den Quantenteil von Simons Algorithmus haben wir bereits analysiert. Wir interessieren uns an dieser Stelle für den *klassischen* Anteil. Aus den Schritten 1.-6. erhalten wir zwei linear unabhängige Lösungen,

$$z_1 = (1,1,0)^T, \quad z_2 = (0,1,0)^T.$$

Simons- & Shors Algorithmus

D.h. das Gleichungsssytem ist

$$\begin{aligned} t_2 \oplus t_1 &= 0 \\ t_1 &= 0. \end{aligned}$$

Für dessen Lösung gilt $t_2 = t_1 = 0$, wobei t_0 nicht festgelegt ist.

Da in **Schritt 7** nach einer Lösung ungleich dem Nullvektor gesucht wird und t_0 frei gewählt werden kann, wird $t_0 = 1$ gesetzt. Also ist $s = (0, 0, 1)^T$.

Simons- & Shors Algorithmus

In **Schritt 8** wird geprüft ob $f(0) = f(s)$ gilt:

$$f(0) = f(0,0,0) = (0,0,0) = f(0,0,1) = f(s).$$

Also hat $f(x_1, x_1, x_0) = (x_2, x_1, 0)$ die Periode $s = (0,0,1)^T$.

Anmerkung: Simons Algorithmus war das erste Beispiel für einen Quantenalgorithmus der jedem klassischen deterministischen Algorithmus für die gleiche Aufgabe exponentiell überlegen ist.

Simons- & Shors Algorithmus

Simons Algorithmus ermittelt die Periode einer Funktion $f : \{0,1\}^n \to \{0,1\}^n$.

Im Faktorisierungsverfahren (Schritt 3) zum RSA-Verfahren war die Bestimmung der Periode der Funktion

$$f(x) = a^x \pmod{n}$$

ein wesentlicher Bestandteil. FFT_n löst dies in einer Laufzeit von $O(n \log n)$.

Wir beschränken den Modulus auf eine 2er Potenz, d.h. $N = 2^n$, für $f(x) = a^x \pmod{N}$.

Simons- & Shors Algorithmus

Der Algorithmus von Shor löst das Problem „ähnlich“ wie wir es bereits bei Simons Algorithmus gesehen haben:

Eingabe: Eine Funktion $f : \{0, ..., N-1\} \to \{0, ..., n-1\}$ mit Periode p, d.h. $f(x+p) = f(x) \forall x \in \{0, ..., N-1\}$.
f steht mittels $U_f : |a\rangle|b\rangle \mapsto |a\rangle|b \oplus f(a)\rangle$ zur Verfügung.

Ausgabe:
1. Fall p teilt N: Erhalte ein ganzzahliges Vielfaches y von $\frac{N}{p}$, also $y = j\frac{N}{p}$.
2. Fall Sonst: Erhalte eine Zahl y „nahe“ einem ganzzahligen Vielfachen von $\frac{N}{p}$.

Simons- & Shors Algorithmus

Shors Algorithmus

1. $R = |a\rangle|b\rangle \leftarrow |0...0\rangle|0...0\rangle$
2. Wende Hadamard-Transformation H_N auf $|a\rangle$ an:

$$R \leftarrow H_N|a\rangle|b\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\{0,...,N-1\}} |x\rangle|0...0\rangle.$$

3. Wende U_f an:

$$R \leftarrow U_f R = \frac{1}{\sqrt{N}} \sum_{x\in\{0,...,N-1\}} |x\rangle|f(x)\rangle.$$

Simons- & Shors Algorithmus

Shors Algorithmus

4. Messe das Register $|b\rangle$
5. Wende QFT_N auf $|a\rangle$ an: $|a\rangle \leftarrow QFT_N|a\rangle$
6. Messe das Register $|a\rangle$ und gib das Ergebnis aus.
7. Ermittle aus y und N mit Hilfe von Kettenbrüchen die Periode p.

Die Schritte 1.-6. bilden den Quantenanteil von Shors Algorithmus. In Schritt 7 ist, wie schon bei Simons Algorithmus, eine *klassische* Nachbearbeitung notwendig. Statt Hadamard-Transformation wurde QFT verwendet.

Simons- & Shors Algorithmus

Beispiel: Bestimme die Periode von

$$f(x) = 7^x \pmod{15}$$

wobei die Zahl 7 zufällig unter den zu 15 teilerfremden Zahlen $\{2, 4, 7, 8, 11, 13, 14\}$ gewählt wurde. Wir wählen auch $N = 16$.

Nach **Schritt 2** befindet sich das Register im Zustand

$$|\phi_2\rangle = \frac{1}{\sqrt{16}} \sum_{x=0}^{15} |x\rangle|0\rangle = \frac{1}{4} \left(|0\rangle|0\rangle + |1\rangle|0\rangle + \ldots + |15\rangle|0\rangle\right).$$

Simons- & Shors Algorithmus

Anwendung von $U_f : |a\rangle|b\rangle \mapsto |a\rangle|b \oplus f(a)\rangle$ in **Schritt 3** führt zu

$$\begin{aligned}
|\phi_3\rangle &= \frac{1}{4}\sum_{x=0}^{15} |x\rangle|7^x \pmod{15}\rangle \\
&= \frac{1}{4}\left(|0\rangle|7^0 \pmod{15}\rangle + |1\rangle|7 \pmod{15}\rangle + \ldots\right) \\
&= \frac{1}{4}\left(|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle + |4\rangle|1\rangle + |5\rangle|7\rangle + \ldots\right)
\end{aligned}$$

An dieser Stelle sieht man das die Periode $p = 4$ ist, denn $f(0) = f(4)$ (und weiter $f(5) = f(1)$).

Simons- & Shors Algorithmus

Da im nächsten Schritt gemessen werden soll, sortieren wir noch nach $|b\rangle$ und erhalten

$$\begin{aligned}|\phi_3\rangle = &\frac{1}{4}\left(|0\rangle + |4\rangle + |8\rangle + |12\rangle\right)|1\rangle + \\ &+ \frac{1}{4}\left(|1\rangle + |5\rangle + |9\rangle + |13\rangle\right)|7\rangle + \\ &+ \frac{1}{4}\left(|2\rangle + |6\rangle + |10\rangle + |14\rangle\right)|4\rangle + \\ &+ \frac{1}{4}\left(|3\rangle + |7\rangle + |11\rangle + |15\rangle\right)|13\rangle.\end{aligned}$$

Simons- & Shors Algorithmus

Eine Messung in **Schritt 4** liefert eine *Zeile* (vgl. vorherige Folie) der Superposition, etwa

$$\frac{1}{2}\left(|1\rangle + |5\rangle + |9\rangle + |13\rangle\right)|7\rangle.$$

Die Periode kann man zwar am Zustand sehen, bei einer Messung erhalten wir jedoch keine Information über p selbst.

Simons- & Shors Algorithmus

Darum wird in **Schritt 5** QFT auf $|a\rangle$ angewendet, mit dem Ergebnis

$$\mathrm{QFT}_{16}\,\frac{1}{2}\left(|0\rangle+|4\rangle+|8\rangle+|12\rangle\right)=\frac{1}{2}\left(|0\rangle+|4\rangle+|8\rangle+|12\rangle\right)$$

$$\mathrm{QFT}_{16}\,\frac{1}{2}\left(|1\rangle+|5\rangle+|9\rangle+|13\rangle\right)=\frac{1}{2}\left(|0\rangle+i|4\rangle-|8\rangle-i|12\rangle\right)$$

$$\mathrm{QFT}_{16}\,\frac{1}{2}\left(|2\rangle+|6\rangle+|10\rangle+|14\rangle\right)=\frac{1}{2}\left(|0\rangle-|4\rangle+|8\rangle-|12\rangle\right)$$

$$\mathrm{QFT}_{16}\,\frac{1}{2}\left(|3\rangle+|7\rangle+|11\rangle+|15\rangle\right)=\frac{1}{2}\left(|0\rangle-i|4\rangle-|8\rangle+i|12\rangle\right)$$

Simons- & Shors Algorithmus

Jede QFT liefert eine Superposition von Vielfachen der (unbekannten) Periode $p = 4$. Dadurch macht QFT die gesuchte Periode durch Messung zugänglich und eine Messung in **Schritt 5** ergibt einen Wert y aus

$$\{0, 4, 8, 12\rangle = \left\{ \frac{jN}{p} \mid j = 0, 1, 2, 3 \right\}.$$

Simons- & Shors Algorithmus

Dabei treten verschiedene Möglichkeiten auf, die wir exemplarisch untersuchen:
Angenommen es wurde $y = 4$ gemessen und j und p haben keine gemeinsamen Teiler. Das ist der Fall für $j = 1$ und $j = 3$ (und die unbekannte Periode $p = 4$). Mit

$$\frac{1}{4} = \frac{4}{16} = \frac{y}{N} = \frac{j}{p}$$

kann für $j = 1$ die Periode direkt abgelesen werden.

Simons- & Shors Algorithmus

Wurde $y = 12$ gemessen, liefert dieser Ansatz

$$\frac{3}{4} = \frac{12}{16} = \frac{y}{N} = \frac{j}{p}$$

und für $j = 3$ kann die Periode direkt abgelesen werden.

Für eine Messung von $y = 0$ oder $y = 8$ kann die Periode jedoch nicht korrekt bestimmt werden.

Aus zahlentheoretischen Gründen ist es allerdings wahrscheinlich das aus dem gerade betrachteten Ansatz die Periode bestimmt werden kann.

Simons- & Shors Algorithmus

Wir wissen: $f(x) = 7^x \pmod{15}$ hat die Periode $p = 4$.

Zurück zu unserem Ausgangsproblem: Wir wollen $n = 15$ faktorisieren (und verwenden den Faktorisierungsalgorithmus).

Wähle zufällig $a \in \{2, ..., 14\}$.

1. Möglichkeit: Es wird zufällig ein echter Teiler a gewählt (Wahrscheinlichkeit dafür ist $2/13$).
Angenommen $a = 3$. Dann ergibt $ggT(a, n) = ggT(3, 15) = 3$ einen echten Teiler von 15 und der Algorithmus endet.
Analog: $a = 5$ liefert den echten Teiler 5.

Simons- & Shors Algorithmus

2. Möglichkeit: Es wird kein echter Teiler gewählt (Wahrscheinlichkeit dafür ist $^{11}/_{13}$).
Angenommen es wurde $a = 7$ gewählt. Dann ist die Periode p von $f(x) = 7^x (\text{mod } 15)$ zu bestimmen.
Wir wissen: $p = 4$. Das ist eine gerade Zahl (d.h. kein Abbruch des Algorithmus) und wir betrachten

$$ggT(a^{p/2} + 1, n) = ggT(7^2 + 1, 15) = ggT(50, 15) = 5$$

bzw.

$$ggT(a^{p/2} - 1, n) = ggT(7^2 - 1, 15) = ggT(48, 15) = 3.$$

Damit wurde $n = 15 = 3 \cdot 5$ fakorisiert.

Simons- & Shors Algorithmus

Der Algorithmus von Shor hat insbesondere Konsequenzen für die „klassische“ asymmetrische Kryptographie, denn er löst

- das Faktorisierungsproblem (RSA), bzw.
- das Problem des diskreten Logarithmus (Diffie-Hellmann)

in polynomieller Laufzeit.

Das motiviert

- Post-Quantum Kryptographie, sowie
- Quantum-Key Distribution.

Inhaltsverzeichnis

QKD - Ziel und Grundannahme

Annahme: Es existiert ein störungsfreier Quantenkanal zwischen Alice und Bob.

Auswirkungen von Störungen / Rauschen auf einen Quantenkanal betrachten wir im Abschnitt *Quantenfehlerkorrektur*.

Ziel: Alice und Bob sollen eine Folge zufälliger und geheimer Bits erhalten (Schlüsselaustausch). Diese können dann für das klassische One-Time Pad verwendet werden.

Das BB84-Protokoll

Vorgehen: Wir erzeugen zunächst ein einzelnes Bit. Das Schlüsselaustauschprotokoll wird dann aus einer Folge solcher Schritte zusammengesetzt

BB84 - Teil 1

Alice verschickt ein Qubit $|x\rangle$.

1. Erzeuge ein zufälliges Bit a.
   Versetze das Qubit in den Zustand $|x\rangle \leftarrow |a\rangle$.
2. Erzeuge ein zweites klassisches Bit a'.
   Ist $a' = 1$, wende Hadamard-Transformation an: $|x\rangle \leftarrow H|x\rangle$.
3. Sende $|x\rangle$ an Bob.

Das BB84-Protokoll

Wie zuvor:

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

Mit Wahrscheinlichkeit von jeweils $1/4$ befindet sich $|x\rangle$ nach Schritt 3 im Zustand $|0\rangle, |1\rangle, |+\rangle, |-\rangle$; genauer

	$a' = 0$	$a' = 1$
$a = 0$	$\|0\rangle$	$\|+\rangle$
$a = 1$	$\|1\rangle$	$\|-\rangle$

Das BB84-Protokoll

Das Zufallsbit a soll an Bob übermittelt werden, wobei a' bestimmt bzgl. welcher Basis a dargestellt wird. Bezeichne

$$B = \{|0\rangle, |1\rangle\}, \quad B' = \{|+\rangle, |-\rangle\}.$$

Ist $|x\rangle$ an Bob übermittelt worden, muss er messen um etwas über a herauszufinden. Da Bob nicht weiß bzgl. welcher Basis, wird die Wahl der Basis dem Zufall überlassen.

Das BB84-Protokoll

BB84 - Teil 2

Bob misst Alice' Qubit. Ist $|x\rangle$ das von Alice erzeugte und übermittelte Qubit, dann

1. Erzeuge ein zufälliges Bit b'.
   Ist $b' = 0$ miss $|x\rangle$ bzgl. $B = \{|0\rangle, |1\rangle\}$.
   Ist $b' = 1$ miss $|x\rangle$ bzgl. $B' = \{|+\rangle, |-\rangle\}$.
2. Teile Alice über den klassischen Kanal mit bzgl. welcher Basis gemessen wurde.

Das BB84-Protokoll

Ist b das Ergebnis der Messung, dann gilt

- $b = 0$, falls $|x\rangle$ vor der Messung im Zustand
  - $|0\rangle$ war und bzgl. B gemessen wurde, oder
  - $|+\rangle$ war und bzgl. B' gemessen wurde.
- $b = 1$, falls $|x\rangle$ vor der Messung im Zustand
  - $|1\rangle$ war und bzgl. B gemessen wurde, oder
  - $|-\rangle$ war und bzgl. B' gemessen wurde.

Übung: Zeige: Ist $b' = a'$, so ist auch $b = a$.

Das BB84-Protokoll

Also: Haben Alice und Bob die gleiche Basis gewählt, dann sind sie auch im Besitz desselben zufälligen Bits. Andernfalls hat Bob nichts über das Bit von Alice erfahren.
Misst Bob bzgl. der *falschen* Basis, treten beide möglichen Ergebnisse mit gleicher Wahrscheinlichkeit auf.

Übung: Zeige: Ist $b' \neq a'$, dann sind die Ergebnisse $b = a$ und $b \neq a$ gleich wahrscheinlich.

Das BB84-Protokoll

Alice und Bob verwenden die Bits a und b nur dann für den Schlüssel, wenn sie beide die gleiche Basis verwendet haben.

BB84 - Teil 3

Alice und Bob entscheiden, ob das Zufallsbit verwendbar ist: Alice teilt Bob mit, ob sie die gleiche Basis verwendet haben. Ist das nicht der Fall, nutzen sie das Ergebnis nicht.

Alice und Bob verwenden in der Hälfte der Fälle die gleiche Basis.

Das BB84-Protokoll

Zusammenfassung:

a	0	0	1	1	0	0	1	1
a'	0	1	0	1	0	1	0	1
$\|x\rangle$	$\|0\rangle$	$\|+\rangle$	$\|1\rangle$	$\|-\rangle$	$\|0\rangle$	$\|+\rangle$	$\|1\rangle$	$\|-\rangle$
b'	0	0	0	0	1	1	1	1
b	0	Z	1	Z	Z	0	Z	1
Schlüssel	0	–	1	–	–	0	–	1

Z bezeichnet hier ein Zufallsbit, dessen Werte 0 oder 1 jeweils mit Wahrscheinlichkeit $1/2$ angenommen werden. Dieses wird nicht in die Schlüsselerzeugung aufgenommen.

Das BB84-Protokoll

Perspektivenwechsel: Welche Möglichkeiten hat Eve?

Eve soll nichts über der Schlüssel, d.h. das Bit a, erfahren.

Nach Kerckhoffs' Prinzip müssen wir davon ausgehen, dass

- Eve weiß welches Verfahren verwendet wird,
- Eve Zugriff auf den Quantenkanal hat, und
- Eve Zugriff auf den klassischen Kanal hat.

Also: Eve kennt a' und b' *nachdem* Bob gemessen hat.

Frage: Kann Eve mit diesem Wissen a bestimmen?

Das BB84-Protokoll

Eve kann $|x\rangle$ nicht kopieren (No-Cloning).

Alternativ: Eve misst das Qubit $|x\rangle$ und

- schickt es an Bob weiter, oder
- erzeugt entsprechend dem Messergebnis ein neues Qubit und schickt dieses an Bob weiter, so als käme es von Alice.

Wahl der Messbasis von Eve

Eve trifft die Wahl ihrer Messbasis zufällig (wie auch Bob):

1. Eve wählt zufällig ein Bit e'.
2. Ist $e' = 0$, misst Eve bzgl. der Basis B. Sonst verwendet sie Basis B'.

Das BB84-Protokoll

Beispiel: $|x\rangle = |+\rangle$, d.h. $a = 0$, $a' = 1$. Es gibt zwei Möglichkeiten:

- Eve wählt $e' = 1$, also $B' = \{|+\rangle, |-\rangle\}$ und das Ergebnis ist $|+\rangle$. Eve folgert $a = 0$ und das Qubit $|+\rangle$ wurde nicht verändert.
- Andernfalls wählt Eve $e' = 0$ und verwendet $B = \{|0\rangle, |1\rangle\}$. Jeweils mit Wahrscheinlichkeit $1/2$ wird $|0\rangle$ oder $|1\rangle$ gemessen und Eve erfährt nichts über a.
  Mehr noch: $|x\rangle$ wurde durch die Messung verändert.

Das BB84-Protokoll

Alice und Bob verwenden das getauschte Bit nur im Fall $a' = b'$. Eves Wahl von e' ist davon unabhängig, d.h. mit Wahrscheinlichkeit $1/2$ hat e' den gleichen Wert und Eve erfährt durch *BB84 - Teil 3* ob dies der Fall ist (sonst ist ihre Messung ein wertloses Zufallsbit).

Also: Eve erfährt die Hälfte des Schlüssels.

Das Verfahren ist so nicht sicher!

Das BB84-Protokoll

Angenommen das übertragene Bit $|x\rangle$ wurde bzgl. derselben Basis gemessen, d.h. es gilt $a' = b'$ und deshalb wird $a = b$ erwartet.

BB84 - Teil 4

Alice und Bob opfern ein Bit um Eve zu entlarven.

1. Alice und Bob tauschen a und b über den klassischen Kanal.
2. Stimmen die Ergebnisse nicht überein, wird die Schlüsselerzeugung abgebrochen und die bereits übertragenen Schlüsselbits nicht verwendet.
3. Stimmen die Ergebnisse überein, wird die Schlüsselübertragung fortgesetzt. Das zur Kontrolle ausgetauschte Bit wird nicht weiter verwendet.

Das BB84-Protokoll

Frage: Wie stehen die Chancen Eve zu bemerken?

Vorausgesetzt $a' = b'$ gibt es zwei Möglichkeiten:

- Eve hat die Basis korrekt geraten, also $a' = b' = e'$. Stimmen a und b überein, dann wird Eve nicht bemerkt.
- Eve hat die falsche Basis geraten. Dann ergibt ihre Messung ein Zufallsbit und mit Wahrscheinlichkeit $1/2$ gilt $a \neq b$. Dabei wird Eve mit Wahrscheinlichkeit $1/4$ entdeckt.

Das BB84-Protokoll

Zusammenfassung:

a	0	0	1	1	0	0	1	1
a'	0	1	0	1	0	1	0	1
b'	0	1	0	1	0	1	0	1
e'	0	0	0	0	1	1	1	1
b	0	Z	1	Z	Z	0	Z	1

Im Fall Z wird Eves Zugriff mit Wahrscheinlichkeit $1/2$ aufgedeckt.

Das BB84-Protokoll

BB84-Protokoll zur Schlüsselerzeugung

1. Alice erzeugt Zufallsbits $a_1, ..., a_m$ und $a'_1, ..., a'_m$.
2. Alice führt für $i = 1, ..., m$ die Schritte durch:
   - kodiere a_i als $|0\rangle$, bzw. $|1\rangle$ falls $a'_i = 0$
   - kodiere a_i als $|+\rangle$, bzw. $|-\rangle$ falls $a'_i = 1$
   - sende das entstandene Qubit an Bob
3. Bob erzeugt Zufallsbits $b'_1, ..., b'_m$. Das i-te Qubit von Alice misst er
   - in der Basis $B = \{|0\rangle, |1\rangle\}$ falls $b'_i = 0$
   - in der Basis $B' = \{|+\rangle, |-\rangle\}$ falls $b'_i = 1$

   und speichere das Ergebnis als b_i.

Das BB84-Protokoll

BB84-Protokoll zur Schlüsselerzeugung

4. Alice und Bob vergleichen für $i = 1, ..., m$ die Bits a'_i und b'_i über einen klassischen Kanal.
   Ist $a'_i \neq b'_i$ werden a_i bzw. b_i nicht weiter verwendet.
5. Alice und Bob tauschen k der nicht gelöschten Bits a_i, b_i aus und ermitteln die Fehlerrate, definiert als die Anzahl der sich unterscheidenden Bits dividiert durch k.
   Ist die Fehlerrate zu hoch, verwenden sie die erzeugten Bits nicht, da der Verdacht auf einen Angriff durch Eve besteht.

Das BB84-Protokoll

Nach unserer Analyse erlaubt das BB84-Protokoll eine sichere Schlüsselverteilung, wenn

- Eve *nur* die zuvor betrachtete Lauschstrategie verwendet (gleich dazu mehr),
- der Quantenkanal perfekt ist (realitätsfremd, also Fehlerkorrekurverfahren).

Lauschstrategie – Verschränkungsangriff

Bisherige Lauschstrategie von Eve: Qubits messen und weiterschicken, denn

- Qubits können nicht kopiert werden, und
- das speichern von Qubits ist aufwendig und (bisher) nur für kurze Zeitspannen möglich.

Aber Eves technische Fähigkeiten können sich weiterentwickeln, d.h. es müssen auch Angriffe berücksichtigt werden die (bisher) nur theoretisch möglich sind: *Verschränkungsangriffe*.

Lauschstrategie – Verschränkungsangriff

Idee: Eve ist im Besitz von Qubits $|e_1\rangle, |e_2\rangle, \ldots$ und wendet unitäre Transformation auf den Quantenkanal an, um so viel Information wie möglich von den gesendeten Qubits auf ihre eigenen Qubits zu übertragen. Dabei will Eve weiterhin möglichst unentdeckt bleiben.

Wir verwenden CNOT um Qubits miteinander zu verschränken und betrachten folgendes **Beispiel**:

Ansatz: Eve verschränkt eines ihrer Qubits mit einem gesendeten Qubit. Dann wartet sie bis Alice und Bob ihre Basen vergleichen und nutzt dies für die Wahl ihrer eigenen Messbasis.

Lauschstrategie – Verschränkungsangriff

Fall 1: Alice sendet $|x\rangle = |1\rangle$. Eve wendet CNOT auf $|x\rangle$ und ihr eigenes Qubit $|e\rangle = |0\rangle$ an und erhält als Ergebnis $|x\rangle|e\rangle = |1\rangle|1\rangle$. Dann wartet Eve bis Bob gemessen hat und belauscht welche Basis er Alice mitteilt. Dabei gilt:

- Misst Bob in der richtigen Basis B, dann auch Eve und Eve erfährt das Schlüsselbit $a = b = 1$.
- Hat Bob die falsche Basis verwendet wird das Bit von Alice und Bob verworfen, also auch von Eve.

Lauschstrategie – Verschränkungsangriff

Fall 2: Alice sendet $|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ und Eve erhält nach Anwendung von CNOT das Ergebnis $|x\rangle|e\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$.

2.1 Bob verwendet B (falsche Basis). Dann beobachtet er $|0\rangle$ und $|1\rangle$ jeweils mit Wahrscheinlichkeit $1/2$. Eve misst auch bzgl. B und erhält ebenfalls $|0\rangle$ und $|1\rangle$ (da die Qubits verschränkt waren).
Jedoch verwenden Alice und Bob das Bit nicht, da die Basis von Bob und Alice nicht übereinstimmen.

Lauschstrategie – Verschränkungsangriff

2.2 Bob verwendet B' (richtige Basis). Dann wird Eve in der Hälfte der Fälle bemerkt werden: Wir wissen $|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = H|1\rangle$ und H ist selbstinvers, d.h. Messung bzgl. B' liefert das gleiche Ergebnis wie Anwendung von H mit anschließender Messung bzgl B:

$$\begin{aligned}(H|x\rangle)|e\rangle &= \frac{1}{\sqrt{2}}\left((H|0\rangle)|0\rangle - (H|1\rangle)|1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) - \frac{1}{\sqrt{2}}(|01\rangle - |11\rangle)\right) \\ &= \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right).\end{aligned}$$

Lauschstrategie – Verschränkungsangriff

2.2 Bob misst das erste Qubit und das Ergebnis stimmt in der Hälfte der Fälle nicht mit Alice überein (andere Wahrscheinlichkeitsverteilung), obwohl beide die Basis B' verwendet haben.

Insgesamt lässt sich feststellen: Verschränkt Eve ihr Qubit mit dem über den Quantenkanal gesendeten Qubit, dann beeinflusst sie das Ergebnis der Messung von Bob.

Lauschstrategie – Verschränkungsangriff

Die Situation ist für Eve also nicht besser, als wenn sie das übertragene Qubit direkt messen würde.

Allgemeinere Angriffsmodelle zeigen:

Je mehr Information Eve erhält, desto stärker wird das gesendete Qubit $|x\rangle$ verändert.

Problematisch bleiben man-in-the-middle Angriffe. Dafür gibt es aber Authentisierungsverfahren.

Das E91-Protokoll (Idee)

Wir beenden den Ausblick in Richtung Quantum Key Distribution mit einer (naiven) Darstelleung des E91-Protokolls.

Idee: Verwende verschränkte Qubits.

Wir wissen:

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

ist ein Paar verschränkter Qubits. Angenommen Alice hat Φ^+ erzeugt und sendet eines der Qubits an Bob. Messen beide ihre Hälfte des Qubits erhalten sie mit Wahrscheinlichkeit $1/2$ dasselbe Ergebnis 0 oder 1.

Das E91-Protokoll (Idee)

D.h. nach der Messung besitzen Alice und Bob jeweils das gleiche Bit *ohne* das dies auf bisherige Weise ausgetauscht werden musste.

Eves Ansatz: Wende auf ein eigenes Qubits $|e\rangle = |0\rangle$ und das von Alice zur Erzeugung von Verschränkung CNOT an. Das liefert

$$\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

und egal in welcher Reihenfolge nun gemessen wird, Eves Ergebnis ist das gleiche, d.h. Eve erfährt den Schlüssel.

Das E91-Protokoll (Idee)

Problematisch für Eve: Ihr verschränktes Qubit beeinflusst wieder das Ergebnis der Messung (wie zuvor bei BB84).

Eves Problem ist Bestandteil des Schlüsselverteilungsprotokolls von *Ekert*. Darin bestimmen Alice und Bob wieder zufällig, welche Messbasis sie verwenden.

Wir skizzieren das Verfahren an dieser Stelle nur, da wir nicht alle notwendigen Vorkenntnisse (*CHSH-Ungleichung*) für eine detaillierte Betrachtung zur Verfügung haben.

Das E91-Protokoll (Idee)

Das E91-Protokoll (Idee)

Alice und Bob steht eine Auswahl an verschiedenen Messverfahren zur Verfügung.

1. Erzeuge ein Paar von Qubits im Zustand $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Ein Bit erhält Alice, das andere erhält Bob.
2. Alice wählt zufällig ein Messverfahren und misst ihr Bit.
3. Bob wählt zufällig ein Messverfahren und misst sein Bit.

Das E91-Protokoll (Idee)

Das E91-Protokoll (Idee)

4. Alice und Bob teilen sich ihr jeweiliges Messverfahren mit.
   - Haben sie das gleiche Verfahren gewählt, dann wird das Messergebnis ein Teil des Schlüssels.
   - Haben sie verschiedene Verfahren gewählt, verwenden sie ihre Ergebnisse zum Test einer *Bedingung C*. Ist diese Bedingung nicht erfüllt, wird der gesamte bisher ausgetauschte Schlüssel verworfen.

Die hier nicht näher betrachtete *Bedingung C* ist abhängig von der CHSH-Ungleichung. Die Bedingung dient zur Feststellung ob Alice und Bobs Qubits maximal verschränkt sind oder ob diese Eigenschaft durch Eves Einwirkung abgeschwächt wurde.

Inhaltsverzeichnis

QEC - Ziel und Grundannahme

Im Abschnitt QKD wurde ein störungsfreier Quantenkanal zwischen Alice und Bob angenommen. Das ist unrealistisch.

Wir betrachten nun Auswirkungen von Störungen / Rauschen auf einen Quantenkanal (*Quantenfehlerkorrektur*).