5 – Quantencomputer

Wie geht es weiter?
Grundsatzfrage

Es gibt also Probleme, die – wahrscheinlich – nicht effizient korrekt gelöst werden können.

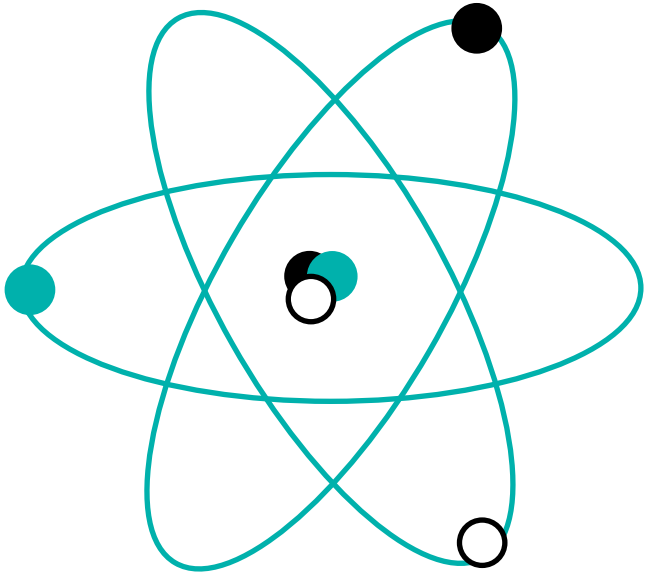
Und jetzt?

Quantencomputer
Allgemeines

Quantencomputer sind – theoretisch – in der Lage bestimmte Probleme effizienter als herkömmliche Computer zu lösen.

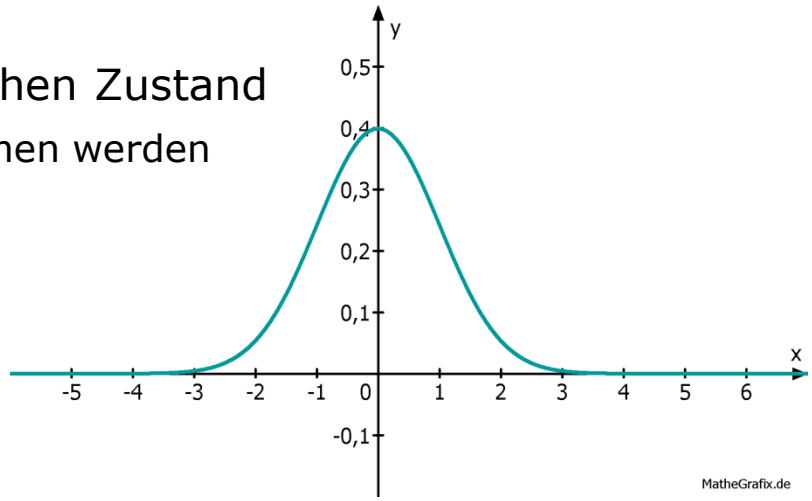
Die Funktionsweise von Quantencomputern beruht auf quantenphysikalischen Effekten

- Superposition
- Quantenverschränkung

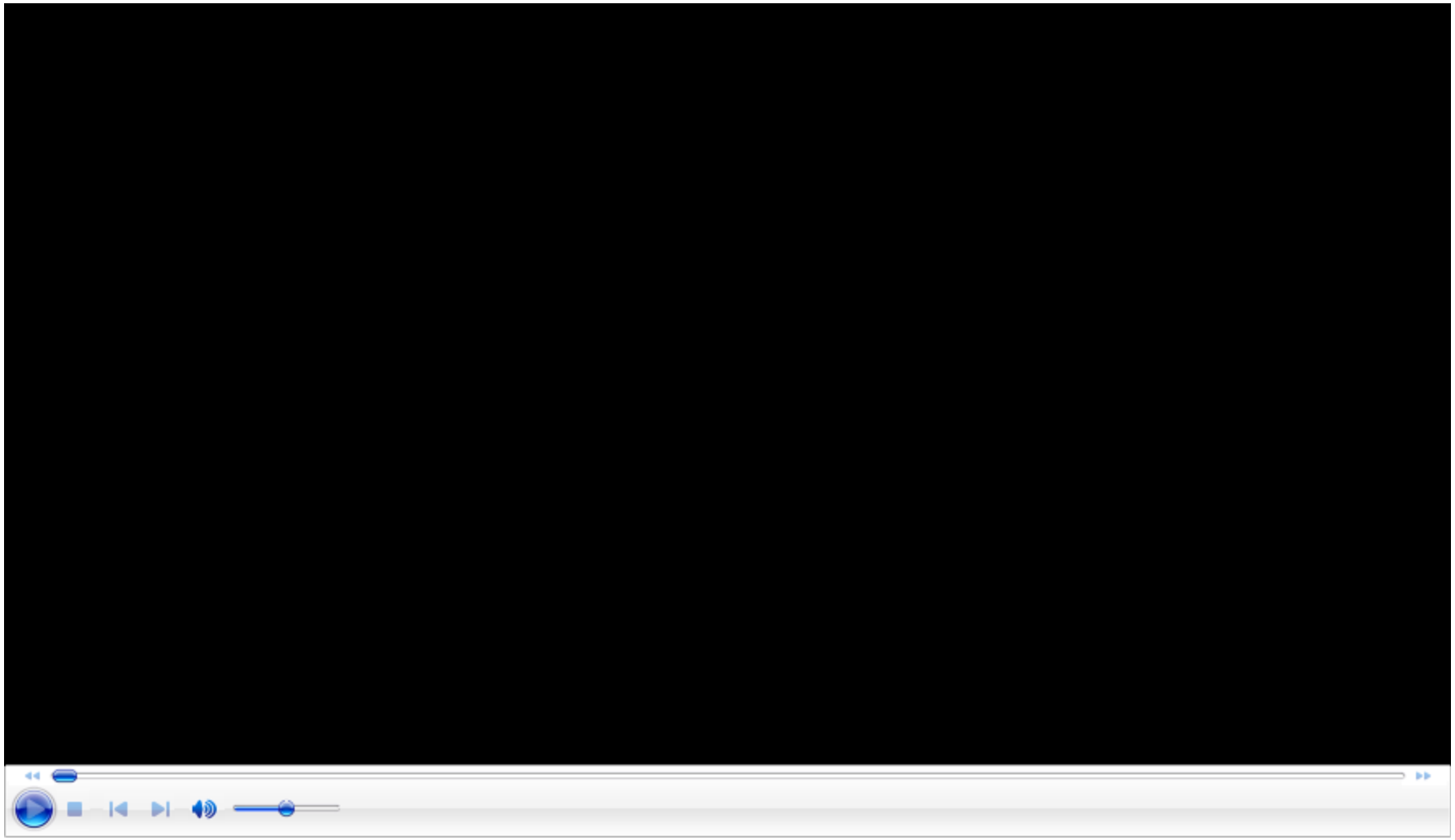


Normalerweise haben Objekte definierte und messbare Eigenschaften, aber …

- Quantenteilchen existieren in allen ihren möglichen Zuständen gleichzeitig
  - > Es befindet sich in Superposition
- Das Ergebnis der Messung einer Eigenschaft ist über eine Wahrscheinlichkeitsverteilung gegeben
  - > Ort
  - > Geschwindigkeit
  - > …
- Messung "zwingt" Teilchen in einen möglichen Zustand
  - > Superpositionen können nicht wahrgenommen werden



Quantencomputer
Schrödingers Katze



Quantencomputer
Qubits

Ein Qubit kann, wie normale Bits auch, die Zustände „0" und „1" haben, aber …

- Ein Qubit kann auch in Superposition dieser beiden Zustände sein

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle$$

  > c_0 und c_1 sind komplexe Zahlen und werden Amplituden genannt
  > Für die Amplituden muss gelten: $|c_0|^2 + |c_1|^2 = 1$

- Eine alternative Darstellung eines Qubits ist aus der linearen Algebra motiviert

$$\psi = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = c_0 * \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_1 * \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- Die Messung des Wertes eines Qubits ergibt immer entweder „0" oder „1"
  > Superpositionen können nicht erfasst werden

Auf ein Qubit können, analog zu klassischen Bits, Operationen angewendet werden

- Die Operationen werden durch unitäre Matrizen dargestellt
  - > Für eine unitäre Matrix U gilt $U * U = I$
  - > Die Länge des Vektors – und damit die Gesamtwahrscheinlichkeit der Zustände des Qubits – bleibt gleich

Beispiel:

- Sei ψ_0 der Zustand eines Qubits und U eine Operation auf dem Qubit
- Dann berechnet sich der Folgezustand ψ_1 des Qubits wie folgt

$$\psi_1 = U|\psi_0\rangle = U * \psi_0$$

Analog zu normalen Registern aus normalen Bits gibt es Quantenregistern aus Qubits.

- Ein Quantenregister kann in Superposition über alle seine Bits sein
  - > Der theoretische Informationsgehalt eines Quantenregisters wächst exponentiell mit der Anzahl der Bits
- Ein Quantenregister R aus n Qubits sieht wie folgt aus
  - > $R = |x_n \dots x_2 x_1\rangle$
  - > Die Darstellung der einzelnen Qubits werden dabei miteinander multipliziert

Beispiel:

- Es seien x_0 und x_1 zwei Qubits mit
  - > $|x_0\rangle = a_0 * |0\rangle + a_1 * |1\rangle$
  - > $|x_1\rangle = b_0 * |0\rangle + b_1 * |1\rangle$
- Die Multiplikation ergibt dann
  - > $R = a_0 b_0 |0\rangle|0\rangle + a_1 b_0 |0\rangle|1\rangle + a_0 b_1 |1\rangle|0\rangle + a_1 b_1 |1\rangle|1\rangle$
- Die Faktoren vor den jeweiligen Zuständen des Registers werden jeweils zu einem r_i zusammengefasst

Ein Quantenregister kann alternativ auch in Vektorschreibweise dargestellt werden

$$\vec{r} = \begin{pmatrix} r_0 \\ r_1 \\ \dots \\ r_{2^{n-1}} \end{pmatrix}$$

- Die Basisvektoren $|i\rangle$ entsprechen den einzelnen Zuständen des Quantenregisters

$$|i\rangle \in \{0,1\}^n$$

- Für die Länge des Vektors, und damit für die Wahrscheinlichkeitsamplituden, gilt demnach

$$R = \sum_{i=0}^{2^{n-1}} |r_i|^2 = 1$$

Eine Operation auf einem Quantenregister wird als Quantengatter bezeichnet

- Die Operation wird in Form einer unitären Matrix auf das Quantenregister angewendet
- Werden mehrere Quantengatter nacheinander auf ein Quantenregister angewendet bezeichnet man dies als Quantenschaltkreis

Die Operationen können auch auf einer Superposition eines Quantenregisters angewendet.

- Dadurch werden alle Basiszustände der Superposition in einen neuen Zustand überführt
- Die Anwendung eines Quantengatters / Quantenschaltkreises auf ein Quantenregister in Superposition wird Quantenparallelismus genannt

Beispiel: Die Hadamard-Transformation

- Die zugehörige Matrix sieht wie folgt aus:

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

- Es seien außerdem zwei Qubits $\psi_0 = |0\rangle$ und $\xi_0 = |1\rangle$ gegeben

- Dann funktioniert die Anwendung der Hadamard-Transformation wie folgt

$$\psi_1 = H|\psi_0\rangle = H * \begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}} * \begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}} * (|0\rangle + |1\rangle)$$

$$\xi_1 = H|\xi_0\rangle = H * \begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}} * \begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}} * (|0\rangle - |1\rangle)$$

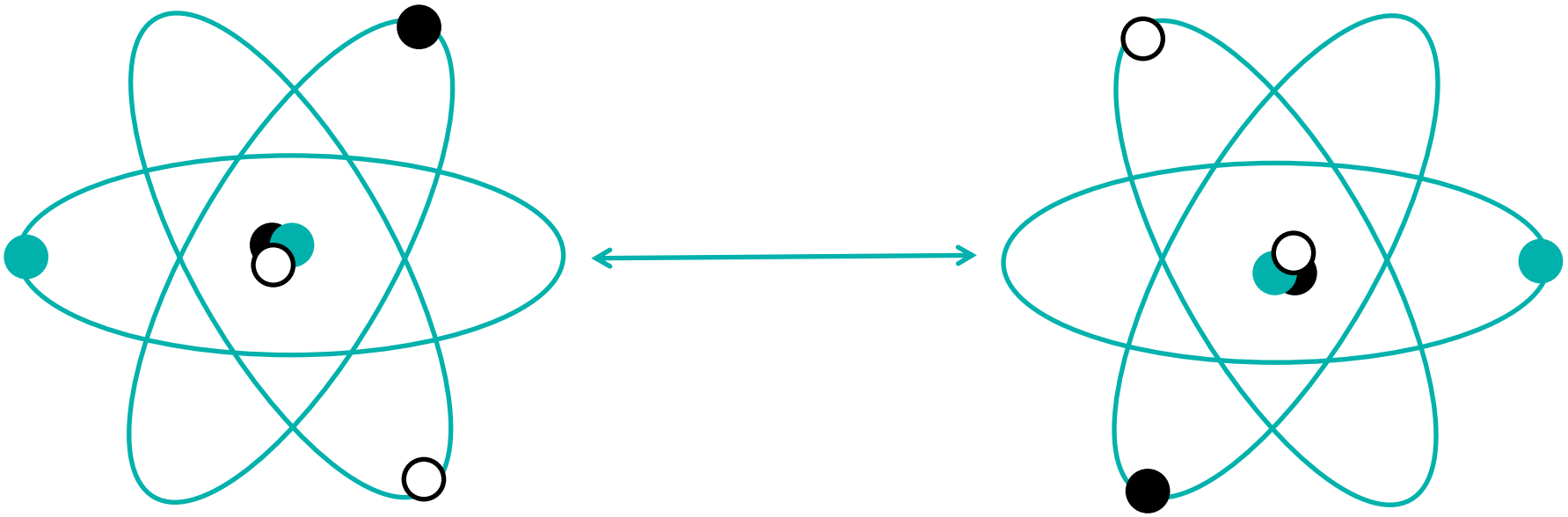
- Die Qubits befinden sich nach der Transformation in einer Superposition der möglichen Zustände, in welchem beide Zustände gleich wahrscheinlich sind

- Da H eine unitäre Matrix ist, ergibt die erneute Anwendung der Hadamard-Transformation wieder die Originalzustände der Qubits

Quantencomputer
Quantenverschränkung

Zwei oder mehr Quantenteilchen können verschränkt sein

- Verschränkung drückt eine Zusammengehörigkeit aus
- Die einzelnen Teilchen können nicht mehr einzeln beschrieben werden, sondern nur das verschränkte Teilchensystem als Ganzes
- Änderungen an einem der Teilchen wirken sich auf alle anderen Teilchen des verschränkten Systems aus



Quantencomputer
Quantenverschränkung – Beispiel

Gegeben sei ein Quantenregister R mit 2 Qubits

- Das Register ist wie folgt initialisiert

$$\mathbf{R} = |\mathbf{00}\rangle$$

- Auf das erste Qubit des Initialzustandes wird die Hadamard-Transformation angewandt, sodass sich folgender Registerzustand ergibt

$$\boldsymbol{R} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}(|\mathbf{00}\rangle + |\mathbf{10}\rangle)$$

- Auf diesen Zustand wie das CNOT Quantengatter angewendet, welches die Quantenversion des normalen XOR ist. Die zugehörige Matrix sieht wie folgt aus

$$\begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{pmatrix}$$

- Das CNOT Quantengatter negiert das zweite Bit des Quantenregisters dann und nur dann, wenn das erste Bit 1 ist.

Quantencomputer
Quantenverschränkung – Beispiel

- Nach Anwendung des CNOT Quantengatters befindet sich das Quantenregister in folgendem Zustand

$$\boldsymbol{R} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}(|\mathbf{00}\rangle + |\mathbf{11}\rangle)$$

- Das Quantenregister befindet sich nur in einem verschränkten Zustand
  - > Eine Messung des ersten Qubits zwingt auch das zweite Qubit in einen bestimmten Zustand.
  - > Das zweite Qubit hat – auch ohne Messung am zweiten Qubit – immer denselben Wert, den das erste Qubit bei einer Messung annimmt.

Grovers Algorithmus ist ein Suchalgorithmus für unsortierte Datenmengen.

- Probabilistischer Algorithmus mit Laufzeitverhalten $O(\sqrt{n})$

- Anwendung auf andere Probleme bei entsprechender Formulierung des Problems möglich
  - > NP-vollständige Probleme
  - > Optimierungsprobleme

- Mathematische Formulierung
  - > Es gibt einen Suchraum R mit N Elementen $\{0, 1, \ldots, N-1\}$
  - > Es gibt in R genau einen Elementwert $\hat{x}$, welcher gesucht werden soll
  - > Die Elementen in R sind in Binärrepräsentation mit maximal $n = \log_2 N$ Bits dargestellt
  - > Es gibt einen Suchfunktion $f: \{0,1\}^n \rightarrow \{0,1\}$

$$f(x) = \begin{cases} 1, & x = \hat{x} \\ 0, & sonst \end{cases}$$

Grundidee: Starte mit einer Superposition aller Elemente des Suchraums und verstärke iterativ die Amplitude des gesuchten Elements („amplitude amplification“)

- Es gibt ein Quantenorakel O, welches für ein $|x\rangle \in N$ angibt, ob es das gesuchte Element ist
- Für jedes $|x\rangle$ wird zusätzlich ein $|y\rangle$ gespeichert, welches das Ergebnis des Quantenorakels enthält
- Das Quantenorakel funktioniert wie folgt: $O|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$

In der Anwendung funktioniert die Grover Iteration dann wie folgt

- $|y\rangle$ wird mit $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ initialisiert, sodass die Anwendung des Quantenorakels die Basiszustände vertauscht, sofern $f(x) = 1$
- Das Vorzeichen der Wahrscheinlichkeitsamplitude der gesuchten Elemente wird durch die Anwendung des Quantenorakels also negiert, sodass die gesuchten Element nun „markiert" sind

Aber: Der Betrag der Amplitude hat sich nicht geändert, somit ist die Chance bei einer Messung das gesuchte Element zu erhalten nicht gestiegen.

Damit die Wahrscheinlichkeitsamplitude der markierten Elemente größer wird, ist noch ein weitere Schritt notwendig.

- Es wird der Mittelwert aller Amplituden berechnet und jede Amplitude an diesem Mittelwert gespiegelt.

Quantencomputer
Grovers Algorithmus - Beispiel

Gegeben sei ein Suchraum $R = \{x_0 = 00, x_1 = 01, x_2 = 10, x_3 = 11\}$

- Gestartet wird mit einer Superposition über alle Basiszustände, also

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

- Angenommen, dass x_2 das Suchkriterium erfüllt, dann ergibt sich nach Anwendung des Quantenorakels folgender Zustand

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

- Der Mittelwert aller Amplituden ist dann

$$\mu = \frac{\frac{1}{2} + \frac{1}{2} - \frac{1}{2} + \frac{1}{2}}{4} = \frac{1}{4}$$

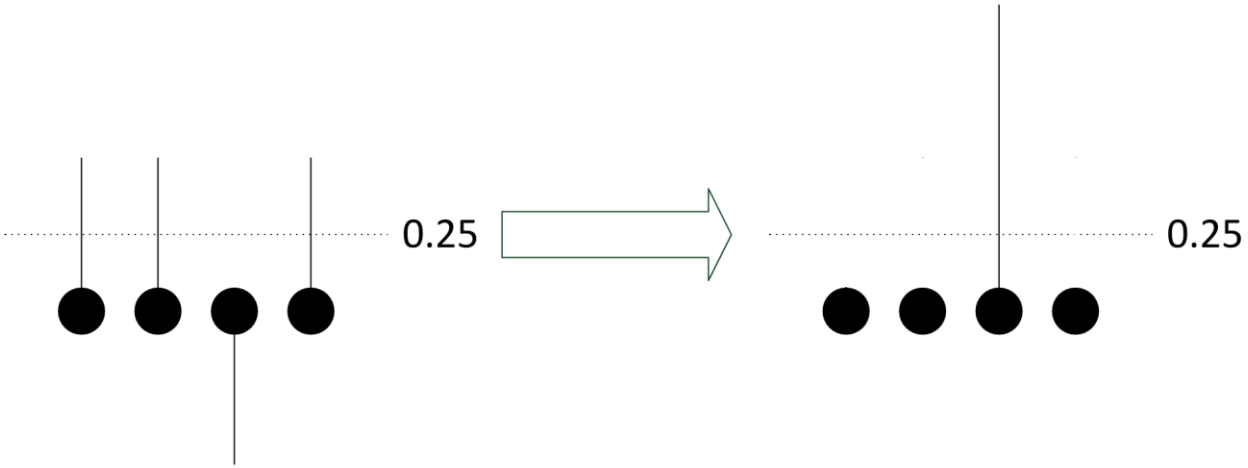
- Das Spiegel an einer Wert erfüllt folgende Gleichung

$$v_{new} = 2 * \mu - v_{old}$$

Quantencomputer
Grovers Algorithmus - Beispiel

- Für das gegebene Beispiel ergeben sich also folgende Wahrscheinlichkeitsamplituden

$$R = 0 * (|00\rangle + |01\rangle + |11\rangle) + 1 * (|10\rangle)$$



- Im vorliegenden Beispiel reicht also eine Anwendung der Grover-Iteration aus, um das gesuchte Element korrekt zu bestimmen

Quantencomputer
Shors Algorithmus

Es gibt keinen Algorithmus, der ganze Zahlen auf einem klassischen Computer effizient faktorisiert, aber...

- Shors Algorithmus faktorisiert einen ganze Zahl auf einem Quantencomputer effizient, d.h. in polynomieller Laufzeit
- In seinem Paper "*Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*" stellt Shor zwei verschiedene Algorithmen vor. Der Algorithmus zur Faktorisierung ganzer Zahlen ist allerdings der bekanntere von Beiden.

Shors Algorithmus löst nicht die Faktorisierung im eigentlichen Sinne, sondern ein Ersatzproblem, auf das die Faktorisierung zurückgeführt werden kann.

Quantencomputer
Shors Algorithmus - Vorüberlegungen

Wenn es darum geht die Faktorisierung einer Zahl N zu bestimmen, dann...

- Gibt es ein x, sodass $x^2 \equiv 1 \, mod \, N$ bzw. $x^2 - 1 \equiv 0 \, mod \, N$

- Wendet man die 3. binomische Formel an, so ergibt sich folgende Gleichung:

$$(x + 1) * (x - 1) \equiv 0 \, mod \, N$$

- Das bedeutet, dass ein Teiler von N der größte gemeinsame Teiler von N und $(x + 1)$ oder $(x - 1)$ sein muss.

- Der größte gemeinsame Teiler kann mit dem euklidischen Algorithmus in polynomieller Laufzeit berechnet werden.

Aber: Welchen Wert hat x?!

Quantencomputer
Shors Algorithmus – Zum Ersatzproblem…

Manche Funktionen sind periodisch …

- Das bedeutet, dass $f(k) = f(k + r) \; \forall \; k$
  - z.B. ist $\sin(k) = \sin(k + 2\pi)$

- Behauptung:
  - **Für ein gegebenes N kann ein x gefunden werden, indem die Periode von $f_a(k) = a^k \; mod \; N$ bestimmt wird.**

- Für ein gewähltes a gibt es ein kleinstes r, sodass $a^r \; mod \; N = 1$

- Dieses r ist die Periode der Funktion, denn…

$$\begin{aligned} a^{k+r} \; mod \; N &= a^k * a^r \; mod \; N \\ &= a^k \; mod \; N \; * a^r \; mod \; N \\ &= a^k \; mod \; N \; * 1 \\ &= a^k \; mod \; N \end{aligned}$$

Quantencomputer
Shors Algorithmus – … und zurück

Hat man r für einen bestimmten Anwendungsfall gefunden, so ist $x = a^{\frac{r}{2}}$

- Das funktioniert offensichtlich nicht, wenn r ungerade ist oder $\left|a^{\frac{r}{2}}\right| = 1$
- In diesem Fall muss ein anderes a ausgewählt und erneut das zugehörige r bestimmt werden
- Die Wahrscheinlichkeit ein neues a auswählen zu müssen beträgt etwa $p = 0.5$. Das bedeutet dass Shors Algorithmus ein Monte-Carlo-Algorithmus ist.

Quantencomputer
Shors Algorithmus – Beispiel

Geben sind $N = 15$ und $a = 7$

- Daraus ergibt sich folgende Funktionsauswertung

k	0	1	2	3	4	5	...
$f_7(k)$	1	7	4	13	1	7	...

- Damit ergibt sich eine Periode von $\mathbf{4}$ und damit $x = 7^{\frac{4}{2}} = 7^2 = 49$

- Damit muss der größte gemeinsame Teiler von 15 und $x + 1 = 50$ bzw. $x - 1 = 48$ ein Primfaktor von N sein

- Berechnung des ggT:
  - > $ggT(15, 48) = 3$
  - > $ggT(15, 50) = 5$

Quantencomputer
Shors Algorithmus – Ablauf

1. Wähle eine Zahl $a < N$

2. Prüfe, ob a zufällig ein Teiler von N ist

3. Berechne die Periode r von $f_a(x)$ (← Das ist der Quantenteil des Algorithmus)
   > Die genaue Funktionsweise würde den Rahmen dieser Vorlesung sprengen

4. Wenn r alle Anforderungen erfüllt, berechne einen Primfaktor von N, ansonsten fahre mit Schritt 1 fort

5. Der größte gemeinsame Teiler von $x^{\frac{r}{2}}$ und N ist sehr wahrscheinlich ein Primfaktor von N, in seltenen Fällen allerdings selbst wieder eine zusammengesetzte Zahl
   > z.B. für RSA nicht relevant