Quantenrechner

Ideen der Informatik

Kurt Mehlhorn

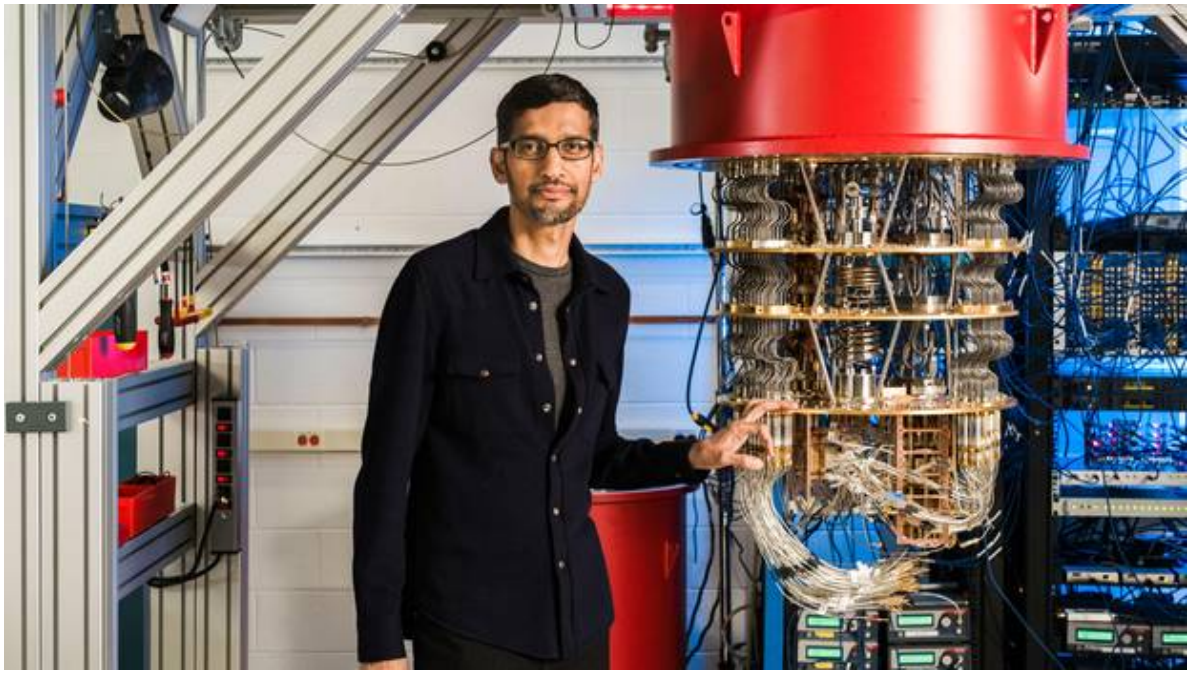
max planck institut
informatik

SIC Saarland
Informatics Campus

Übersicht

- Was sind Quantenrechner?
- Vorteile von Quantenrechnern
- Quantenüberlegenheit: Das Experiment von Google
- Qbits und Überlagerungen
- Störungen, Fehlerkorrektur und Beobachtbarkeit
- Grovers Algorithmus, ein Beispiel für Quantenüberlegenheit
- Technische Realisierung
- Zusammenfassung

Stern Oktober 2019



Quantencomputer: Google gelingt die Computer-Revolution

Seit Wochen kursierten Gerüchte - nun ist es offiziell: Zum ersten Mal hat ein Quantencomputer eine Aufgabe erledigt, an der herkömmliche Rechner scheitern würden. Das behauptet zumindest Google. Konkurrent IBM zweifelt das Ergebnis allerdings an.

Quantencomputer

Heutige Rechner beruhen auf klassischer Physik (Physik von Newton). Quantenphysik (Planck, Einstein, ...) ist nötig zum Verständnis der Welt auf atomarer Skala. Quantenphysik erlaubt mehr Phänomene als die klassische Physik.

Computer, die Quanteneffekte ausnützen, sind potentiell mächtiger als klassische Computer.

Sie können nicht mehr berechnen als klassische Computer, aber sie können unter Umständen manche Probleme schneller lösen (Quantenüberlegenheit, Quantum Supremacy).

Beispiele von Kandidaten für Quantenüberlegenheit:

- Simulation der Quantenphysik
- Faktorisierung: Gegeben eine Zahl finde ihre Darstellung als Produkt von Primzahlen. Schon für 2000-stellige Zahlen ist das auf klassischen Computern praktisch nicht lösbar (= dauert mehrere Jahre). Bedroht Teile der Kryptographie.

Das Experiment von Google

- Die Maschine erzeugt zufällige Bitstrings der Länge 52 (nicht uniforme Verteilung). Nutzt dazu Quanteneffekte. Sie erzeugt 1 Mio Strings in ein paar Millisekunden.
- Die Quantenphysik der halben Maschine kann man auf einem klassischen Supercomputer simulieren. Die Simulation bestätigt, dass die Maschine Quanteneffekte nutzt.
- Extrapolation der Supercomputer-Rechenzeit: Simulation der ganzen Maschine braucht mehrere Jahre (laut Google).
- Google: damit ist Quantenüberlegenheit gezeigt.
- Einwände:
  - Ist die Simulation auf einem klassischen Rechner bestmöglich? IBM hat Zeit für Simulation auf Tage verbessert.
  - Gelöste Aufgabe ist nicht interessant.

Vorteile von Quantenrechnern

Heutige Rechner beruhen auf klassischer Physik. Quantenphysik erlaubt Rechner, die für **manche** Probleme potentiell schneller sind. Realisierung steht noch am Anfang.

Problem	klassisch	Quantenrechner
Sortieren, schnellste Wege, ...	keine Änderung	
Faktorisieren	kein polynomieller Alg. bekannt	Polynomzeitalgorithmus (Peter Schor)
Simulation von Quantenphysik	kein polynomieller Alg. bekannt	Polynomzeitalgorithmus
Suchen in ungeordneter Datenbank	kein sublinearer Alg. möglich	$\sqrt{\text{Größe}}$ möglich (Lov Grover)
Sichere Datenübertragung	nur unter Annahmen, z.B. Faktorisieren ist schwer	möglich, ohne jegl. Annahmen
Asymm. Kryptogr.	nur unter Annahmen, z.B. Faktorisieren ist ...	viele Verfahren werden unsicher
Symm. Kryptogr.	da ändert sich nichts	

Was macht Quantenrechner so mächtig?

Klassisch: Ein Register mit n Bits ist in genau einem von $N = 2^n$ möglichen Zuständen. Wir identifizieren die möglichen Zustände mit den Zahlen 0 bis $N - 1$.

Quantenrechner: Ein Quantenregister mit n Qbits (Quantenbits) kann gleichzeitig ein bisschen in jedem der N möglichen Zustände sein.

Das Register ist in einer Überlagerung (Superposition) der N möglichen Zustände: mit Gewicht w_0 im Zustand 0, mit Gewicht w_1 im Zustand 1, ..., mit Gewicht w_{N-1} im Zustand $N - 1$.

Rechnungen operieren auf diesen Überlagerungen und wirken **parallel** auf allen 2^n reinen Zuständen. Parallelität ist aber nicht beliebig.

$n = 20$, klassisch: in **genau einem** von 10^6 Zuständen;
Quanten: **ein bisschen in jedem** der 10^6.

Überlagerungen

Zustand eines Quantenregisters mit n Qbits ist ein Vektor

$$(w_0, w_1, \ldots, w_{N-1})$$

von $N = 2^n$ komplexen Zahlen mit Norm 1, d.h., $\sum_i |w_i|^2 = 1$.
Die Gewichte (Amplituden) sind also komplexe Zahlen.

Beispiel für 2 Qbits: $\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$.

Komplexe Zahlen sind eine Obermenge der reellen Zahlen. Eine komplexe Zahl ist ein Paar von reellen Zahlen, das man schreibt als $a + bi$, wobei $i = \sqrt{-1}$ und a und b reelle Zahlen. In den Beispielen kommen nur reelle Zahlen vor (mit einer Ausnahme).

$$(a + bi) + (c + di) = (a + c) + (b + d)i \qquad \text{Addition}$$

$$(a + bi) \cdot (c + di) = (ac - bd) + (ad + bc)i \qquad \text{Multiplikation}$$

$$|w|^2 = a^2 + b^2 \qquad \text{Betrag, } w = a + bi$$

$n = 20$, klassisch: Zustand ist **EINE ganze Zahl zwischen 0 und** 10^6
Quanten: Zustand ist ein **Vektor von** 10^6 **komplexen Zahlen.**

Rechnen

Zustand eines Quantenregisters mit n Qbits ist ein Vektor

$$(w_0, w_1, \ldots, w_{N-1})$$

von $N = 2^n$ komplexen Zahlen mit Norm 1, d.h., $\sum_i |w_i|^2 = 1$.

Grundoperationen (Gatter): Quantenphysik erlaubt im Prinzip jede Operation, die jeden Vektor der Norm 1 eineindeutig in einen Vektor der Norm 1 überführt (unitäre Transformationen).

$w \mapsto Uw$, wobei U eine komplexe Matrix mit Determinante ± 1.

$\begin{pmatrix} 1/2 & 1/2 \\ -1 & 1 \end{pmatrix}$ und $\begin{pmatrix} i/2 & 1/2 \\ 1 & i \end{pmatrix}$ haben Determinante ± 1.

Technisch realisieren kann man Gatter, die auf einem oder zwei Qbits operieren. Es gibt eine kleine universelle Basis von Gattern.

Messungen und Störungen

Quantenrechner rechnen im Verborgenen. Wenn man ein Register inspiziert, dann sieht man einen klassischen Zustand. Man sieht den Zustand i mit Wahrscheinlichkeit $|w_i|^2$.

Störungen (Noise) sind unvermeidbar und beschränken die maximale Tiefe von Berechnungen im Augenblick auf ≤ 50 Schritte.

Fehlerkorrektur ist möglich aber teuer: 10^3 physische Qbits pro perfektem Qbit.

Stand der Kunst = NISQs (noisy intermediate-scale quantum computer): ≤ 100 Qbits, ≤ 50 Rechenzyklen.

Google (2019): 53 Qbits, 30 Zyklen.

Faktorisierung nach Shor braucht für eine Zahl mit 1024 Bits etwa 2300 perfekte Qbits und eine Rechenzeit von mehreren Stunden.

Algorithmen für Quantenrechner

Erfordern eine neue Denkweise, da die Grundoperationen gänzliche andere sind.

Erfahrung im klassischen Algorithmenentwurf hilft wenig.

Ich werde Ihnen den Suchalgorithmus von Grover zeigen.

Aufgabe: Gegeben ist eine Funktion f mit n Argumenten $x_i \in \{0,1\}$ und einem Ausgabebit mit der Eigenschaft, dass es genau ein x^* gibt mit $f(x^*) = 1$. Finde x^*.

Klassisch braucht man dafür Zeit $N = 2^n$ im schlechtesten Fall und Zeit $N/2$ im Mittel. Schneller geht es nicht.
Quantenrechner können die Aufgabe in Zeit $O(\sqrt{N})$ lösen.

Suchen (ohne weitere Information)

Aufgabe: Gegeben ist eine Funktion f mit n booleschen Eingaben in $\{0,1\}$ und einem Ausgabebit mit der Eigenschaft, dass es genau ein x^* gibt mit $f(x^*) = 1$. Finde x^*.

Klassisch: man geht die $N = 2^n$ möglichen Eingaben durch, bis man x^* findet. (Vorlesung Suchen)

Laufzeit: N im schlechtesten Fall, $N/2$ im Mittel.

Besser geht es nicht.

Annahme für Quantenalgorithmus: Man kann Phasen-Inversion bei x^* realisieren, d.h. Vorzeichen der Amplitude von x^* wird geändert, alle anderen Amplituden bleiben gleich.

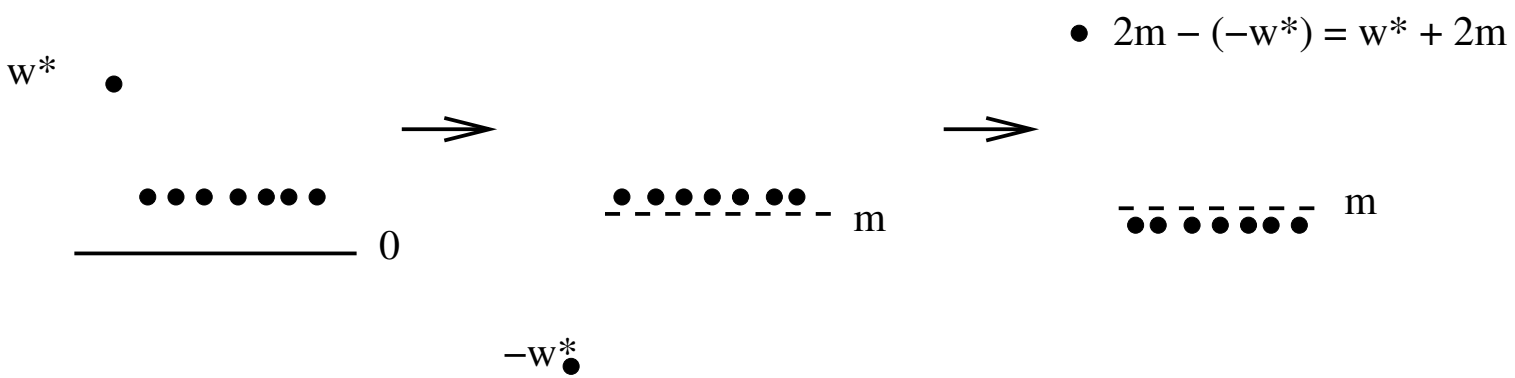
Das ist eine unitäre Transformation. Ich kann nicht beurteilen, für welche f die Transformation technisch realisiert werden kann.

Grovers Quantenalgorithmus braucht nur Zeit $O(\sqrt{N})$.

Der Algorithmus benutzt zwei Operationen; beide Operationen sind unitär und daher im Prinzip möglich. Alle Gewichte werden am Anfang auf $1/\sqrt{N}$ gesetzt. Die Gewichte bleiben reell.

Phasen Inversion bei x^*: Vorzeichen des Gewichts von x^* wird geändert, alle anderen Gewichte bleiben gleich.

Spiegeln am Mittelwert: Für alle i wird w_i ersetzt durch $m + (m - w_i)$, wobei m der Mittelwert aller Gewichte ist.



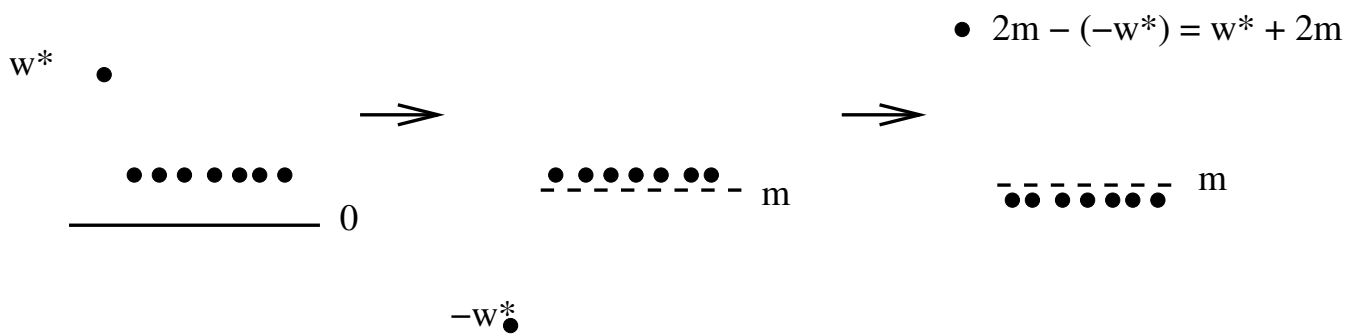
Grovers Quantenalgorithmus braucht nur Zeit $O(\sqrt{N})$.

Initialisierung: $w_i := \frac{1}{\sqrt{N}}$ für alle i.

Wiederhole

- Phasen Inversion, d.h. flippe Vorzeichen des Gewichts von x^*.
- Spiegeln am Mittelwert

bis das Gewicht von x^* über $1/\sqrt{2}$ liegt. Inspektion liefert dann x^* mit Wahrscheinlichkeit $\geq 1/2$.



Das Gewicht von x^* wächst in einer Iteration um das Doppelte des aktuellen Mittelwerts. Der Mittelwert ist am Anfang $1/\sqrt{N}$ und wird im Laufe der Rechnung unwesentlich kleiner. Also wächst das Gewicht von x^* im Wesentlichen um $1/\sqrt{N}$ in jeder Iteration. Also $O(\sqrt{N})$ Iterationen. Vorsicht.

Spiegeln am Mittelwert ist eine unitäre Transformation

Jedes w_i wird abgebildet auf $2m - w_i$, wobei $m = \sum_{1 \leq j \leq n} w_j / n$. Also

$$w_i \mapsto \left(2 \sum_{1 \leq j \leq n} \frac{w_j}{n} \right) - w_i.$$

Die Transformationsmatrix U hat das Element $2/n - 1$ auf der Diagonale und $2/n$ überall sonst. Also

$$U = \begin{pmatrix} 2/n - 1 & 2/n & \dots & 2/n \\ 2/n & 2/n - 1 & \dots & 2/n \\ \vdots & & & \\ 2/n & 2/n & \dots & 2/n - 1 \end{pmatrix}.$$

Und nun die Berechnung der Determinante von U.

Berechnung der Determinante der Transformationsmatrix

$$\det U = \begin{vmatrix} 2/n-1 & 2/n & \dots & 2/n \\ 2/n & 2/n-1 & \dots & 2/n \\ \vdots & & & \\ 2/n & 2/n & \dots & 2/n-1 \end{vmatrix} = \begin{vmatrix} -1 & 0 & \dots & 1 \\ 0 & -1 & \dots & 1 \\ \vdots & & & \\ 2/n & 2/n & \dots & 2/n-1 \end{vmatrix}$$

$$= \begin{vmatrix} -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & & & \\ 2/n & 2/n & \dots & 2/n-1+(n-1)\cdot 2/n \end{vmatrix} = (-1)^{n-1}$$

wobei wir zuerst die letzte Zeile von allen anderen Zeilen abgezogen haben und dann die ersten $n-1$ Spalten zur letzten Spalte addiert haben.

Beachten Sie, dass sich dadurch in der rechten unteren Ecke der Wert $2/n-1+(n-1)\cdot 2/n = 2/n-1+2-2/n = 1$ ergibt.

Nun stehen über der Diagonalen lauter Nullen und auf der Diagonale ist ein Wert $+1$ und alle anderen -1. Also hat die Determinante den Wert $(-1)^{n-1}$.

Stand der technischen Realisierung

Einige Firmen (IBM, Google, Microsoft, Intel, Alibaba) und einige Startups entwicklen Quantencomputer. Universitäten und MPIs arbeiten an den Grundlagen, z.B. Prof. Mauchs, UdS, MPI für Quantenoptik, Helmholtzzentrum Juelich.

Stand der Kunst = NISQs (Noisy intermediate-scale quantum computers): $\leq$ 100 Qbits, $\leq$ 50 Rechenzyklen.

Faktorisieren einer 1024-bit Zahl braucht 2300 perfekte Qbits und mehrere Stunden Rechenzeit. Für ein perfektes Qbit braucht man im Augenblick 1000 physische Qbits.

Haupthindernisse gemäß NAS (National Academy of Science, USA)

- Qubits Cannot Intrinsically Reject Noise.
- Error-Free QC Requires Quantum Error Correction.
- Large Data Inputs Cannot Be Loaded into a QC Efficiently.
- Quantum Algorithm Design Is Challenging.
- Quantum Computers Will Need a New Software Stack.
- The Intermediate State of a Quantum Computer Cannot Be Measured Directly.

Quantenrechner: Zusammenfassung

Heutige Rechner beruhen auf klassischer Physik.
Quantenphysik erlaubt Rechner, die für **manche** Probleme potentiell schneller sind. Realisierung steht noch am Anfang.

Problem	klassisch	Quantenrechner
Sortieren, schnellste Wege, ...	keine Änderung	
Faktorisieren	kein polynomieller Alg. bekannt	Polynomzeitalgorithmus (Peter Schor)
Simulation von Quantenphysik	kein polynomieller Alg. bekannt	Polynomzeitalgorithmus
Suchen in ungeordneter Datenbank	kein sublinearer Alg. möglich	$\sqrt{\text{Größe}}$ möglich (Lov Grover)
Sichere Datenübertragung	nur unter Annahmen, z.B. Faktorisieren ist schwer	möglich, ohne jegl. Annahmen
Asymm. Kryptogr.	nur unter Annahmen, z.B. Faktorisieren ist ...	viele Verfahren werden unsicher
Symm. Kryptogr.	da ändert sich nichts